

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»**

Приладобудівний факультет  
Кафедра інформаційно-вимірювальних технологій

«До захисту допущено»

Завідувач кафедри ІВТ

\_\_\_\_\_ **Володимир ЄРЕМЕНКО**

(підпис)

(ініціали, прізвище)

“    ” \_\_\_\_\_ 2020 р.

**Дипломний проект**

**на здобуття ступеня бакалавра**

за освітньо-професійною програмою

**«Інформаційні вимірювальні технології та системи»**

спеціальності **152 «Метрологія та інформаційно-вимірювальна техніка»**

на тему: **«Система сліпого відновлення зображень за допомогою  
нейронних мереж»**

Виконав :     студент **IV** курсу, групи **ВМ-61-2**

\_\_\_\_\_ **Малиняк Володимир Володимирович**

(прізвище, ім'я, по батькові)

(підпис)

Керівник

**старший викладач Козир Олег Васильович**

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

(підпис)

Консультант

\_\_\_\_\_

(назва розділу)

\_\_\_\_\_

(посада, вчене звання, науковий ступінь, прізвище, ініціали)

\_\_\_\_\_

(підпис)

Рецензент

\_\_\_\_\_

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище ім'я по батькові)

(підпис)

Засвідчую, що у цьому дипломному  
проекті немає запозичень з праць інших  
авторів без відповідних посилань.

Студент \_\_\_\_\_

(підпис)

Київ – 2020 року

**Національний технічний університет України  
«Київський політехнічний інститут  
імені Ігоря Сікорського»**

Факультет (інститут) приладобудівний факультет  
(повна назва)

Кафедра інформаційно-вимірювальних технологій  
(повна назва)

Рівень вищої освіти – перший (бакалаврський)

Спеціальність 152 «Метрологія та інформаційно-вимірювальна техніка»

Освітньо-професійна програма  
«Інформаційні вимірювальні технології та системи»

ЗАТВЕРДЖУЮ

Завідувач кафедри

Володимир ЄРЕМЕНКО

(підпис) (ініціали, прізвище)

«\_\_» \_\_\_\_\_ 20\_\_ р.

**ЗАВДАННЯ**  
**на дипломний проєкт студенту**  
Малиняку Володимиру Володимировичу

(прізвище, ім'я, по батькові)

1. Тема проєкту (роботи) Система сліпого відновлення зображень за допомогою нейронних мереж

керівник проєкту (роботи) Козир Олег Васильович старший викладач ,  
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «\_\_» \_\_\_\_\_ 20\_\_ р. №\_\_

2. Строк подання студентом проєкту (роботи) \_\_\_\_\_

3. Вихідні дані до проєкту (роботи) 1.Дефектні цифрові зображення .

4. Зміст пояснювальної записки (перелік завдань, які потрібно розробити)  
записки ТЗ. Вступ. Загальні положення .Огляд і порівняльний аналіз існуючих методів сліпого відновлення зображень. Поняття нейронної мережі. Розробка системи сліпого відновлення зображення із допомогою нейронних мереж. Експериментальне дослідження і розрахунок похибок .Результати виконаної роботи.

5. Перелік (ілюстративного) графічного матеріалу (з точним зазначенням обов'язкових креслеників, плакатів тощо) Схема структурна. Схема функціональна. Схема алгоритмічна.

6. Консультанти розділів проєкту (роботи)\*

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв

7. Дата видачі завдання 11 лютого 2020 року

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1.	Розробка та узгодження технічного завдання	22.02.2020 р.	
2.	Огляд і аналіз існуючих технічних рішень	05.03.2020 р.	
3.	Проектування структурної схеми	13.03.2020 р.	
4.	Проектування функційної схеми	02.04.2020 р.	
5.	Проектування алгоритмічної схеми	14.04.2020 р.	
6.	Аналіз похибок вимірювальних каналів	14.05.2020 р.	
7.	Оформлення графічних матеріалів	01.06.2020 р.	
8.	Оформлення пояснювальної записки	01.06.2020 р.	
9.	Попередній захист дипломного проєкту	07.06.2020 р.	
10.	Рецензування дипломного проєкту	до 10.06.2020 р.	
11.	Захист дипломного проєкту	до 20.06.2020 р.	

Студент

\_\_\_\_\_

(підпис)

Малиняк Володимир

(ініціали, прізвище)

Керівник проєкту

\_\_\_\_\_

(підпис)

Олег КОЗИР

(ініціали, прізвище)

\_\_\_\_\_

\* Консультантом не може бути зазначено керівника дипломного проєкту (роботи)

## АНОТАЦІЯ

Дипломний проєкт складається зі вступу, шести розділів, висновків, 2 додатків. Загальний об'єм пояснювальної записки становить 79 сторінок, 43 рисунків, перелік посилань.

В ході виконання дипломного проєкту була розроблена система об'єктом розробки якої є модуль програмного забезпечення інформаційної системи сліпого відновлення зображень із допомогою нейронних мереж.

Мета роботи - розробка програмного забезпечення інформаційної системи для відновлення цифрових зображень незалежно від їх походження із допомогою обчислювальних систем типу нейронна мережа. Ключовою особливістю системи є можливість створення необмеженої кількості даних для калібрації систем відновлення зображень на основі симуляції штучних даних із застосуванням реалістичної моделі шуму і моделюванням внутрішніх процесів обробки цифрових камер.

Представлена система може бути використана як окреме ПЗ для візуального покращення окремих фотографій ,так і у складі процесу формування цифрового зображення у камерах , незалежно від їх виду, архітектури, сенсора. Тому, розробка представляє цінність для компаній виробників у вигляді додаткового, націленого на результат підходу у відновленні і покращенні зображень зробленими їхніми пристроями.

Для демонстрації роботи системи були проведені тести у хмарній платформі Google Colab.

## ANNOTATION

Thesis consists of an introduction, six sections, conclusions, 2 appendices. The total volume of the explanatory note is 79 pages, 43 figures, a list of references.

In the course of the thesis a system was developed, the object of development of which is a module of software information system for blind image reconstruction using neural networks.

The purpose of the work - the development of software information system for the recovery of digital images, regardless of their origin, using computer systems such as neural networks. A key feature of the system is the ability to create an unlimited amount of data for calibration of image recovery systems based on simulation of artificial data using a realistic noise model and modeling of internal processing pipeline of digital cameras.

The presented system can be used as a separate software for visual enhancement of individual photos, and as part of the process of forming a digital image in the cameras, regardless of their type, architecture, sensor. Therefore, the development represents a value for the manufacturing companies in the form of an additional, result-oriented approach to restoring and improving the images taken by their devices.

Tests were performed on the Google Colab cloud platform to demonstrate the system's performance.

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ**  
**«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»**

Кафедра інформаційно-вимірювальних технологій

ЗАТВЕРДЖУЮ

Зав. кафедрою ІВТ

\_\_\_\_\_ проф. Володимир ЄРЕМЕНКО

" \_\_\_\_ " \_\_\_\_\_ 2020 р.

**ТЕХНІЧНЕ ЗАВДАННЯ**

**на дипломний проєкт**

**«Система сліпого відновлення зображень за допомогою  
нейронних мереж»**

**ВМ61.030004.001 ТЗ**

УЗГОДЖЕНО:

Керівник дипломного проєкту

Старший викладач \_\_\_\_\_

(Посада)

Козир О.В.

(П.І.П)

" \_\_\_\_ " \_\_\_\_\_ 2020р.

Дипломник:

студент гр. ВМ-61-2

Малиняк В.В.

(П.І.П)

" \_\_\_\_ " \_\_\_\_\_ 2020

р.

Залікова книжка

\_\_\_\_\_ 61203

Київ 2020

## **1 ПРИЗНАЧЕННЯ ТА ГАЛУЗЬ ЗАСТОСУВАННЯ**

Розроблено ПЗ для виконання відновлення цифрових зображень із невідомою інформацією про умови зйомки і незалежно від типу камери і архітектури сенсора.

## **2 ПІДСТАВИ ДЛЯ ВИКОНАННЯ РОЗРОБКИ**

Підставою для розробки є завдання на дипломний проект, видане і затверджене керівником дипломного проекту Козирем О.В. у жовтні 2019р.

## **3 МЕТА ТА ТЕХНІКО-ЕКОНОМІЧНЕ ОБҐРУНТУВАННЯ РОЗРОБКИ**

Метою розробки є створення на сучасній елементній базі автоматизованої системи сліпого відновлення цифрових зображень незалежно від джерела його походження

Використання системи дозволить:

- Моделювати реалістичні данні у необмеженій кількості для калібрування системи сліпого відновлення .
- Відновлювати втрачену візуальну інформацію використовуючи лише дефектне зображення.

## **4 ДЖЕРЕЛА РОЗРОБКИ**

Джерелами розробки є:

- державні та галузеві стандарти;
- науково-технічна література по проектуванню електронного обладнання;
- матеріали з інтернету.

## **5 ТЕХНІЧНІ ВИМОГИ**

5.1 Вимоги, що визначають експлуатаційні характеристики, показники якості та метрологічні характеристики.

5.1.1 Технічні характеристики, метрологічні характеристики і показники якості.

5.1.1.2 Система повинна отримувати данні із камери.

5.1.1.3 Система повинна виконувати обробку отриманих даних.

5.1.1.4 Система повинна створювати цифровий негатив.

5.1.1.5 Система повинна моделювати реалістичний шум .

5.1.1.6 Продуктивність система повинна не залежати від джерела зображення.

5.1.1.7 Система повинна відновлювати цифрове дефектне зображення

5.2 Система забезпечує збір та відображення інформації .

5.3 Вимоги до часу встановлення робочого режиму.

Система повинна забезпечувати в робочих умовах застосування необхідні характеристики після закінчення часу встановлення робочого режиму. Час встановлення робочого режиму має бути необмеженою.

Система повинна допускати тривалість безперервної роботи.

5.4 Вимоги до надійності

5.4.1 Система відноситься до відновлюваного класу виробів.

5.4.2 Середній час напрацювання на відмову має бути не менше 4000 год. згідно з ГОСТ27883

5.4.3 Середній термін служби повинен бути не менше 10 років.

5.4.4 Середній час відновлення повинен бути не більше 40 хв.

5.5 Вимоги до засобів, методів і умов повірки

5.6 Умови проведення повірки повинні відповідати вимогам ГОСТ 8.395.

5.7 Додаткові вимоги

5.8 Система забезпечує обробку та відображення інформації .



## 6 ЕТАПИ РОЗРОБКИ

Етапи розробки, оформлення та узгодження дипломного проекту наведено в табл. 6.1.

Таблиця 6.1 – Етапи розробки дипломного проекту

№ п/п	Найменування етапу	Дата
1	Розробка та узгодження технічного завдання	12.05.2020
2	Огляд і аналіз існуючих технічних рішень	13.05.2020
3	Проектування схеми структурної	15.05.2020
4	Проектування схеми функційної	20.05.2020
5	Проектування схеми алгоритму виконання	22.05.2020
6	Аналіз продуктивності системи	23.05.2020
7	Оформлення графічних матеріалів	02.06.2020
8	Оформлення пояснювальної записки	04.06.2020
9	Попередній захист дипломного проєкту	05.06.2020
11	Рецензування дипломного проєкту	до 10.06.2020
12	Захист дипломного проєкту	до 17.06.2020

Всі ГОСТи, що використовуються в даному ТЗ на ДП, є чинними на території України.

Нумерація	Формат	Позначка			Найменування	Аркуші	№ екз.	Примітки	
1					Альбом 1				
2									
3					Документація загальна				
4					Заново розроблена				
5	A4	BM61.030004.001 ТП			Відомість технічного проекту	1	1		
6	A4	BM61.030004.002 ПЗ			Пояснювальна записка	81	1		
7	A4	BM61.030004.001 ТЗ			Технічне завдання	2	1		
8									
9	A4	BM61.030004.003 ПЕЗ			Система сліпого відновлення зображень за допомогою нейронних мереж				
10					Перелік елементів	1	1		
11									
12					Альбом 2				
13									
14					Графічна документація				
15					Розроблена заново				
16	A1	BM61.030004.001 Е1			Система сліпого відновлення зображень за допомогою нейронних мереж				
17					Схема структурна	1	1		
18									
19	A1	BM61.030004.002 Е2			Система сліпого відновлення зображень за допомогою нейронних мереж				
20					Схема функціональна	1	1		
21									
22	A1	BM61.030004.002 Е3			Архітектура нейромереж				
23					Схема алгоритмічна	1	1		
					BM61.030004.001 ТП				
Зм.	Арк.	№ докум	Підпис	Дата					
Розроб.		Малиняк В.В.			Система сліпого відновлення зображень за допомогою нейронних мереж Відомість технічного проекту	Літ.	Аркуш	Аркуші	
Перев.		Козир О.В.				Т		1	1
Тех.контр.						КПІ ім. Ігоря Сікорського Каф. ІВТ, гр. BM61-2			
Н.контр.		Богомазов С.А.							
Затвердж.		Єременко Н.В.							

# **Пояснювальна записка до дипломного проєкту**

на тему: «Система сліпого відновлення зображень за допомогою нейронних мереж»

Київ – 2020 року

## ЗМІСТ

ВСТУП.....	3
1 ЗАГАЛЬНІ ПОЛОЖЕННЯ .....	4
1.1 Поняття сліпого відновлення.....	4
1.2 Дискретна модель.....	6
1.3 Функція розмиття точки.....	7
1.4 Будова і природа шуму .....	8
1.5 Процес створення цифрового зображення .....	16
1.6 Доцільність створення системи .....	24
2 ОГЛЯД І ПОРІВНЯЛЬНИЙ АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ СЛІПОГО ВІДНОВЛЕННЯ ЗОБРАЖЕНЬ.....	26
2.1 Локальні методи фільтрації .....	26
2.1.1 Фільтр Гауса .....	26
2.1.2 Фільтри околів.....	27
2.1.3 Білатеральні фільтри .....	28
2.2 Методи на основі диференціальних рівнянь з частковими похідними (PDE).....	29
2.2.1 Анізотропне дифузійне рівняння.....	29
2.2.2 Повна варіація .....	29
2.3 Фільтри в частотній області.....	31
2.3.1. Гранична вейвлет обробка .....	32
2.4 Нелокальні методи .....	33
2.4.1 Алгоритм нелокального усереднення.....	33
3 ПОНЯТТЯ НЕЙРОННОЇ МЕРЕЖІ .....	35
3.1 Штучна нейронна мережа .....	35
3.2 Згорткові нейронні мережі.....	36
3.3 Навчання нейронної мережі.....	38
3.4 Функція втрат і оцінка точності.....	41
4 РОЗРОБКА СИСТЕМИ СЛІПОГО ВІДНОВЛЕННЯ ЗОБРАЖЕННЯ ІЗ ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ.....	43
4.1 Підготовка даних для калібрування системи відновлення .....	43
4.2 Розробка структурної схеми .....	58
4.3 Розробка функціональної схеми .....	59

					<b>ВМ61.030004.001 ПЗ</b>		
Змн.	Лист	№ докум.	Підпись	Дата			
Розраб.		Малиняк В.В.			Система сліпого відновлення за допомогою нейронних мереж Пояснювальна записка	Лім.	Аркуш
Перевір.		Козиць О.В.					1
Реценз.						КПІ ім. Ігоря Сікорського ПБФ, гр. ВМ-61-2	
Н. Контр.		Богомазов С.А.					
Затв.		Єременко В.Н.					

4.4 Розробка алгоритму нейронної мережі відновлення зображень.....	62
4.5 Деталі впровадження.....	66
5 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ І РОЗРАХУНОК ПОХИБОК .....	67
6 РЕЗУЛЬТАТИ ВИКОНАНОЇ РОБОТИ .....	71
ВИСНОВКИ.....	76
ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ.....	77
ДОДАТОК А.....	81
ДОДАТОК Б.....	82

## ВСТУП

Ціль процесу відновлення зображень є у отриманні втраченої візуальної інформації внаслідок не ідеальності будови камери, оптичних і погодних перешкод. Це відбувається як наслідок роботи алгоритмів які використовують статистичну інформацію про систему що створила зображення, особливості сцени ,природу і можливість дефектів. На жаль у більшості реальних задач отримати ці данні неможливо або занадто дорого. Наприклад, у таких сферах як астрономія та дистанційне зондування складно статистично змодельовати потрібні нам об'єкти і сцени які ніколи не знімалися – ми не знаєм як і що фільтрувати. У медицині, підвищення якості зображень буває неможливе для реалізації, наприклад збільшення деталізації рентген знімку потребує приросту інтенсивності гамма-променів, що є неприйнятно.

Сліпе відновлення[26] – прекрасна альтернатива для покращення якості зображень без необхідності у комплексних і складних методах калібрування, процес відновлення може бути реалізований із одним лиш дефектним зображенням.

Останні досягнення у сфері нейронних мереж суттєво покращили продуктивність існуючих методів виступаючи як і окремим програмним забезпеченням так і складі камер [29].

# 1 ЗАГАЛЬНІ ПОЛОЖЕННЯ

## 1.1 Поняття сліпого відновлення

Відновлення зображень використовується у таких сферах як астрономія, аерофотозйомка, УЗД, комп'ютерна томографія, камерах відео нагляду, професійних фотоапаратах, смартфонах – тобто всюди де результатом є зображення. Виробники цих пристроїв із року в рік ведуть напружену боротьбу за першість у якості, детальності і реалістичності зображень створених їхніми пристроями. В хід ідуть передові і експериментальні методи, розробка яких ,потребує величезних фінансових затрат і людських ресурсів.

Зображення є джерелом інформації цінність якого неможливо недооцінити. Шум спотворює сигнал і призводить до втрати візуальної інформації зображення: це може бути як і невеликі «косметичні» спотворення ,що не сильно турбують, так і розмиття які не дадуть змогу розпізнати злоякісну пухлину в тілі пацієнта, або ворожу техніку, позиції на дисплеї БПЛА. Сучасному стану систем відображення інформації відповідає лінійна модель деградації [32]:

$$L(x, y) = \iint_{-\infty - \infty}^{\infty \infty} g(x, y; x', y') \cdot f(x', y') dx' dy' + N(x, y) \quad (1.1)$$

де  $g(x, y; x', y')$  — функція розмиття точки (англ. PSF), результуюча імпульсна характеристика каналу прийому-передач(в якому і виникають

					ВМ61.030004.001 ПЗ	Аркуш
Изм.	Лист	№ докум.	Подпись	Дата		4

спотворення образу). В загальному функція залежить від просторових координат - у різних областях зображення присутні різні функції розмиття. Така функція називається просторово-залежною функцією розмиття точки;  $f(x', y')$  — вихідний сигнал (зображення);  $N(x, y)$  — адитивний шум;  $L(x, y)$  — отримуваний спотворений образ;  $x, y$  — відповідні просторові координати.

У випадку коли  $g(x, y; x', y')$  не залежить від положення у просторі вираз (1.1) перетворюється в інтеграл Дюамеля:

$$L(x, y) = \iint_{-\infty-\infty}^{\infty\infty} g(x - x', y - y') \cdot f(x', y') dx' dy' + N(x, y) \quad (1.2)$$

Задачею відновлення є пошук оцінки  $\hat{f}(x, y)$  вихідного сигналу, такої, що вона за деяким, наперед заданим об'єктивним критерієм, «мало» відрізняється від сигналу  $f(x, y)$ . Більш частіше переходять із аналітичного виразу (1) до його запису у дискретного вигляді:

$$l = g * f + n \quad (1.3)$$

де  $*$  - оператор згортки.

Процес відновлення зображення поділяється на два класи, залежно від того, чи присутня інформація природи деградації. Якщо ця деградація відома, тобто відома оцінка функції розмиття точки  $g$  і природа шуму  $n$ , тоді відновлення  $f$  із  $l$  називається відновленням (розгорткою). Якщо, навпроти, опис і оцінка  $g$  і  $n$  невідома процес називається **сліпим відновленням** [19]. Розгортка із  $l$  є інверсною, не правильно поставленою (ill-posed) задачею і деколи сингулярною. Не правильно поставлена задача, це задача у якій



незначна зміна вхідних даних, в даній ситуації  $g$  і  $l$ , приводить до значних змін або повністю інакшому значені виходу, тобто  $f$ . Іншими словами, проблема відновлення зображення означає, що невеликі обмежені збурення в даних можуть призвести до необмежених відхилень в рішенні, і тому вирішення інверсної задачі марне. Проблема називається сингулярною, якщо крім ill-posed властивості не існує унікального рішення для неї. В обох випадках потрібна регуляризація щоб знайти правильне рішення. Підходи до регуляризації процесу відновлення зображень широко класифікуються двома способами: стохастична регуляризація, яка використовує знання коваріаційних матриць та шуму; і детермінована регуляризація, яка вважає, що більшість природних образів є відносно безсильними із обмеженою високочастотною активністю. Хоча стохастична регуляризація широко використовувалася в минулому, маючи важливий внесок у цю сферу наприклад, фільтр Вінера, останнім часом великий акцент робиться на використанні похідних фільтрів із детермінованою регуляризацією.

## 1.2 Дискретна модель

Якщо рахувати модель як дискретну слідує що [32]:

$$l(x, y) = \sum_{x'} \sum_{y'} g(x - x', y - y') f(x', y') + N(x, y)$$

де  $l(x, y), f(x, y), g(x, y)$  дискретні форми отриманого зображення, оригінального зображення і функція розмиття точки відповідно(рис 1.1).

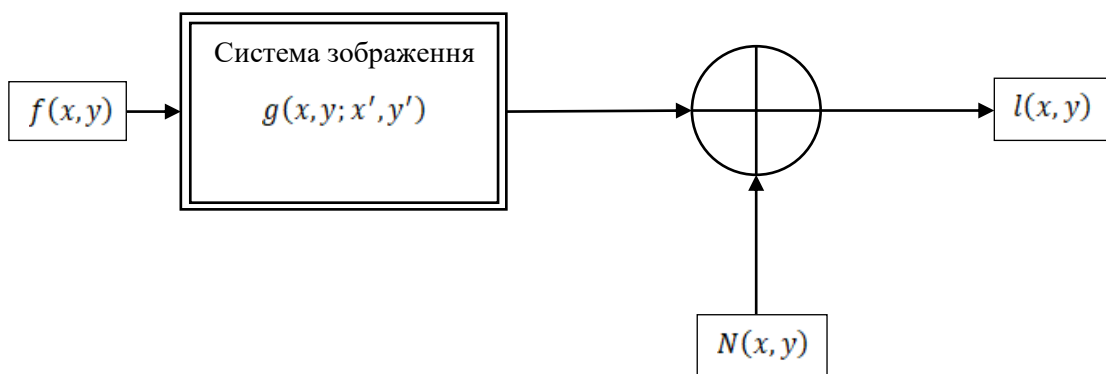


Рисунок 1.1 - Спрощена модель формування зображення

### 1.3 Функція розмиття точки

Функція розмиття точки є загальною абстракцією процесів що призводять до спотворення зображення, її природа зв'язана із будовою камери, оптичними дефектами і природними явищами. Рахується що розмір  $g$  є меншим ніж розмір зображення. Елементи цієї матриці позитивні і реальні ,також рахується що при згортці не міняється середнє значення зображення(рис 1.2).

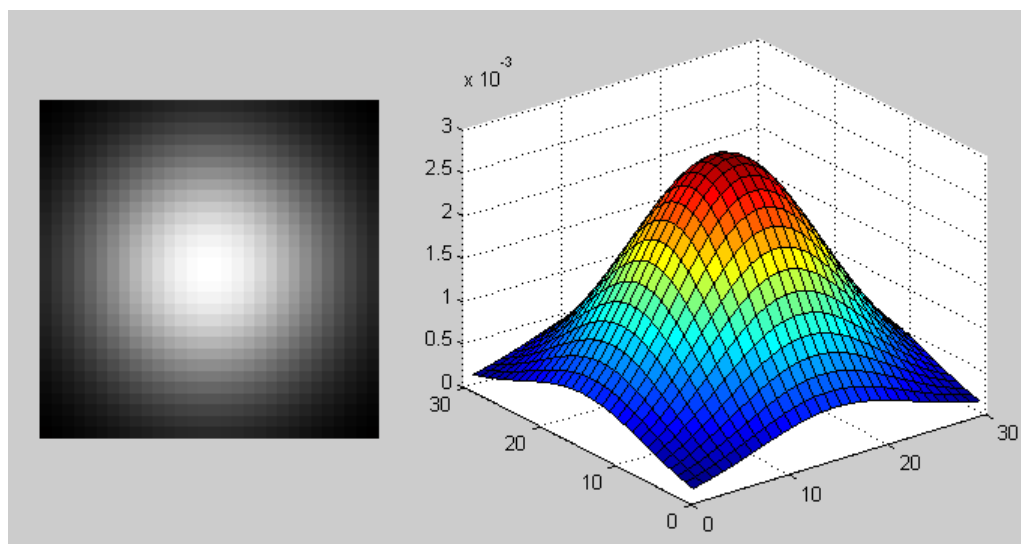


Рисунок 1.2 - Функція розмиття точки по Гаусу

Тобто:

$$g(x, y) \geq 0, \quad 0 \leq x \leq P, \quad 0 \leq y \leq P;$$

$$\sum_x \sum_y g(x, y) = 1 \quad (1.4)$$

#### 1.4 Будова і природа шуму

Шум - одне із ключових понять у сфері обробки зображень, він описує успіх або невдачу якого завгодно процесу передачі, відтворення і відновлення сигналу. Показує наскільки отриманий сигнал можна вважати достовірним і істинним, ілюструє оцінку продуктивності більшості ключових процесів у цифрових камерах[22]. Шум може бути результатом великої кількості джерел включаючи дискретний характер випромінювання, варіацій у чутливості сенсорів, наслідки передачі і зберігання даних, помилки квантування, погодні умови що впливають на процес зйомки і багато інших. До недавнього шум камери моделювали процесом Гауса, тобто

Білим Адитивним Шумом Гауса (AWGN). Функція щільності розподілу шуму по Гаусу (рис 1.3):

$$p_G(z) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(z-\mu)^2}{2\sigma^2}} \quad (1.4)$$

де  $z$  відповідає за рівень сірого,  $\mu$  середнє значення  $z$ ,  $\sigma$  середньоквадратичне відхилення.

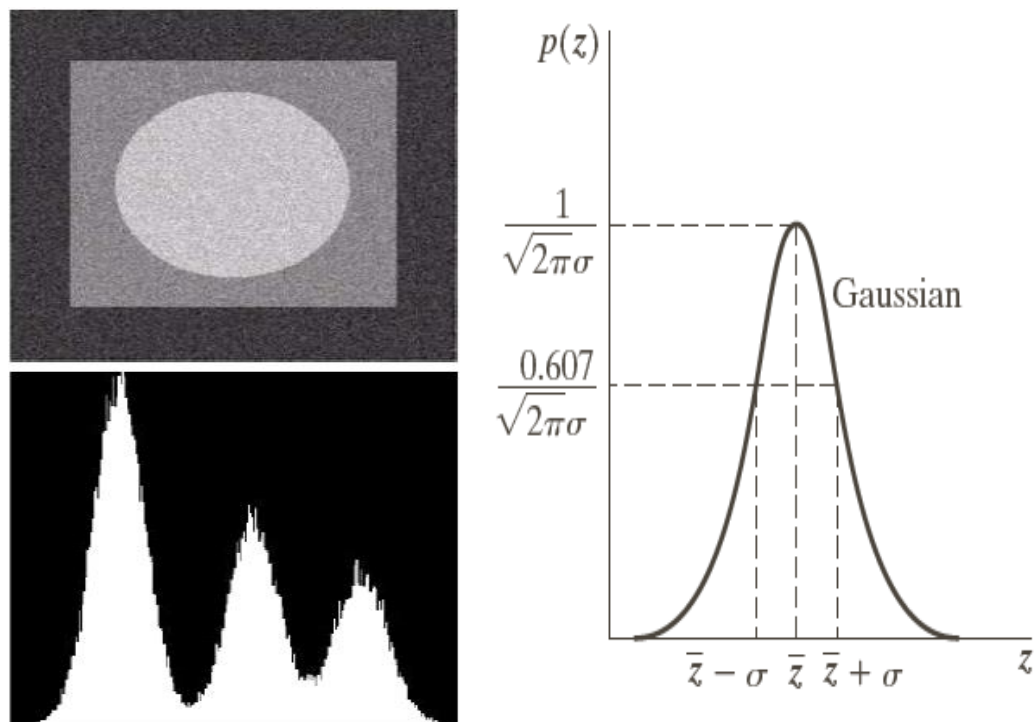


Рисунок 1.3 - Приклад шуму по Гаусу і графік щільності розподілу

Такий підхід призвів до створення прекрасних алгоритмів і методів реконструкції зображень таких як усадка вейвлетів ,метод загальної варіації ,анізотропна дифузія та інші .На жаль, це в своєму роді завдання-іграшка і методи показують низьку якість на реальних зображеннях створених камерами на основі КМОН\ПЗЗ сенсорів. Це виникає внаслідок того що

модель шуму БАШГ майже повністю не відповідає реальності: шум не стаціонарний ,не адитивний гаусовський процес і не білий. В загальному, шум – залежний від сигналу ,комплексний відносно багатьох джерел ,комбінаційний відносно різних лінійних і нелінійних кроків процесу до/після обробки цифрової камери .Для реальних зображень типове:

- Dark current noise – шум який виникає внаслідок матриць які розташовані у камері. В нормальних умовах сенсори реагують на фотони світла ,але часом відбувається їх активація від бродячих електронів що призводить до появи білих точок на зображенню (чимось схоже до шуму солі і перцю)
- Через квантово-механічні аспекти властивості сенсорів що вимірюють світло, сигнал піддається статистичному коливанню із відхиленнями, так званий фотонний або дробовий шум.
- Електричний шум створюється в електроніці камери під час процесу зчитування, оскільки електрони піддаються аналоговому та цифровому перетворенню, посиленню та обробці.
- Під час оцифрування відбувається квантування аналогового сигналу що також приводить до появи шуму, що зветься шумом квантування.
- Сучасні сенсори камер в наслідок розвитку кишенькових камер і смартфонів стали меншими при одночасному рості кількості пікселів. Через це один піксель отримує меншу кількість фотонів за однаковий проміжок часу, внаслідок цього збільшується інтенсивність впливу випадкового адитивного шуму. Для нівелювання логічно було би збільшувати час витримки , що привело до розмиття внаслідок руху.
- Розмиття руху, так званий motion blur, виникає внаслідок руху камери чи предмета зйомки підчас відкритого затвору (витримки) коли світлочутливі датчики акумулюють заряд внаслідок попадання на них фотонів світла. В залежності від інтенсивності руху зображення може

частково деградувати або повністю стати непригідним ,вся інформація буде втрачена

- Також через внутрішні процеси обробки (in-camera processing pipeline) такі як гамма-кореляція , демозаїка і стиснення зображення потерпає від додаткового шуму.

Тому відновлення реальних зображень на основі реалістичних моделей шуму все ще являється складною задачею[24]. Побудова більш складних та реалістичних моделей шуму усуває розрив між все більш мініатюрнішими конструкціями датчиків камери та очікування кінцевих користувачів камери підвищення якості зображення.

Шум зображень можна розділити на До і Після внутрішньої обробки камери. Сире ,так зване Raw зображення, є більш підходящим для моделювання шумів, через очевидну прозорість його побудови і опрацювання. На цьому етапі виникають шуми вплив яких можна спрогнозувати через їхню незворотну природу (темновий, тепловий, фотонний шуми).Але, як було сказано раніше після опрацювання камери (post-processing) містить багато кроків і трансформацій, що ,в свою чергу, трансформує не тільки зображення але і шуми на ньому. Це призводить до не лінійних залежностей що можуть відрізнятись від камери до камери. Вони часто бувають не відомі ,або не задокументовані, моделювання їх впливу не виправдовує затрат.Отже, щоб побудувати узагальнену модель шумів, не залежачи від певного виду камери потрібно працювати із сирими зображенням, технологія побудови яких є зрозуміла і по суті є аксіомою.

Розглянемо гаусовсько-пуассонівську модель шуму[21]. Модель основана на принципі ,що шум залежить від сигналу і моделює на основі Пуассонівського розподілу фотонний шум і Гаусовської моделі для моделювання інших стаціонарних спотворень.

					ВМ61.030004.001 ПЗ	Аркуш
Изм.	Лист	№ докум.	Подпись	Дата		11

Розглядаєм загальну, залежну від сигналу модель спостережень шуму у такій формі :

$$z(x) = y(x) + \sigma(y(x))\xi(x) \quad (1.5)$$

де  $x \in X$  – позиція пікселя у просторі  $x, z : X \rightarrow R$  – спостережений сигнал,  $y : X \rightarrow R$  – не відомий оригінальний сигнал,  $\xi : X \rightarrow R$  – незалежний нульовий середній випадковий шум із середньоквадратичним відхиленням рівним 1,  $\sigma : R \rightarrow R^+$  – функція від  $y$  що відповідає за середньоквадратичне відхилення загального компоненту шуму. Позначаєм математичне сподівання випадкової величини як  $E\{\cdot\}$ , дисперсію як  $var\{\cdot\}$ , і середньоквадратичне відхилення як  $std\{\cdot\} = \sqrt{var\{\cdot\}}$ . Символ  $\sigma$  відповідає за функцію. Із  $E\{\xi(x)\} = 0$  слідує, що  $E\{z(x)\} = y(x)$  тобто, оригінальний сигнал може бути заданий як математичне очікування спостережень шуму. Отже,  $std\{z(x)\} = \sigma(E\{z(x)\})$  середньоквадратичне відхилення шуму є функцією, означеною  $\sigma$ , як математичне очікування зашумленого сигналу. Модель формується із двох незалежних компонентів: Пуасонівський залежний від сигналу компонент  $\eta_p$  і Гаусовський не залежний від сигналу компонент  $\eta_g$ :

$$\sigma(y(x))\xi(x) = \eta_p(y(x)) + \eta_g(x)$$

Їхні розподіли :

					ВМ61.030004.001 ПЗ	Аркуш
Изм.	Лист	№ докум.	Подпись	Дата		12

$$\chi \left( y(x) + \eta_p(y(x)) \right) \sim P(\chi y(x)), \quad \eta_g(x) \sim N(0, b),$$

де  $x > 0, b \geq 0$ , скалярні значення  $P$  і  $N$  відповідають розподілам Пуассона і Гауса. Із елементарних властивостей розподілу Пуассона середнє і дисперсія такі :

$$E \left\{ \chi \left( y(x) + \eta_p(y(x)) \right) \right\} = \text{var} \left\{ \chi \left( y(x) + \eta_p(y(x)) \right) \right\} = \chi y(x).$$

Так як:

$$E \left\{ \chi \left( y(x) + \eta_p(y(x)) \right) \right\} = \chi y(x) + \chi E \{ \eta_p(y(x)) \},$$

$$\chi^2 \text{var} \{ \eta_p(y(x)) \} = \chi y(x)$$

слідуює що:

$$E \{ \eta_p(y(x)) \} = 0 \text{ і } \text{var} \{ \eta_p(y(x)) \} = y(x) / \chi.$$

Тому значення  $\eta_p$  має значення дисперсії залежне від  $y(x)$ ,

$$\text{var} \{ \eta_p(y(x)) \} = a y(x)$$

де  $a = \chi^{-1}$ . Компонент  $\eta_g$  має дисперсію еквівалентну  $b$ . Загальна дисперсія визначається так :

					ВМ61.030004.001 ПЗ	Аркуш
Изм.	Лист	№ докум.	Подпись	Дата		13



$$\sigma^2(y(x)) = ay(x) + b,$$

середньоквадратичне відхилення  $\sigma$  :

$$\sigma(y(x)) = \sqrt{ay(x) + b},$$

$$\sigma(0) = \sqrt{b}, \quad \sigma(1) = \sqrt{a + b};$$

Модель Пуассона-Гауса реалістично підходить для сирого сигналу цифрового сенсора камери [21]. Пуассонівський компонент  $\eta_p$  моделює частину похибок які залежать від сигналу, що зв'язано із квантовою природою. Гаусовський компонент  $\eta_g$  відповідає за похибки які незалежні від сигналу , такі як електричний шум(рис 1.4).

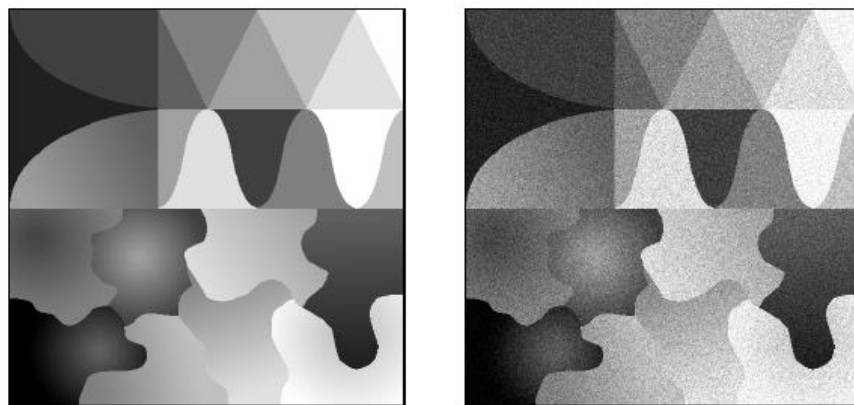


Рисунок 1.4 - Зображення розміром  $512 \times 512$ : оригінальне і деградоване шумом Гауса і Пуассона із параметрами  $x = 100(a = 0,01)$  і  $b = 0.04^2$

- 1) Квантова ефективність: Параметр  $\chi$  компонента  $\eta_p$  моделює зв'язок із квантовою ефективністю сенсора : чим більша кількість фотонів,

необхідних для отримання відповіді датчика(генерація електрона), тим менший  $\chi$ .

- 2) Параметр п'єдесталу: У цифрових сенсорах значення електричного сигналу пікселя завжди додається до якогось базового рівня(рівень п'єдесталу)  $p_0 \in R^+$ . Це призводить до здвигу від нуля у вихідних даних що може буди представлено як зсув аргумента у залежному від сигналу компоненті шуму.

$$z(x) = y(x) + \sigma(y(x) - p_0)\xi(x) = y(x) + \eta_p(y(x) - p_0) + \eta_g(x).$$

- 3) Аналогове підсилення: Моделюємо аналогове підсилення накопленого заряду як мультиплікація сигналу без шуму, шуму Пуассона і частини шуму Гауса масштабуванням  $\theta > 1$ .

- 4) Гетероскедастична нормальна апроксимація : Для спрощення використання ми використовуємо звичну нормальну апроксимацію розподілу Пуассона :

$$P(\lambda) \approx N(\lambda, \lambda)$$

Точність апроксимації збільшується із параметром  $\lambda$ , на практиці для достатньо великих значень  $\lambda$ , Пуасонівський процес може трактуватись як спеціальний гетероскедастичний гаусовський процес. Апроксимації похибок:

$$\sigma(y(x))\xi(x) = \sqrt{ay(x) + b}\xi(x) \approx \eta_h(y(x)),$$

де  $\eta_h \sim N(0, ay(x) + b)$ .

## 1.5 Процес створення цифрового зображення

Щоб зрозуміти як формуються зображення у цифровій камері потрібно розглянути повний цикл, ключові кроки внаслідок яких ми отримуємо фінальне кольорове зображення яке виводиться на дисплей смартфона, фотокамери чи комп'ютера[20].

Людина є трихроматом – сітківка ока має три види рецепторів які реагують на хвилі певної довжини. Разом вони надають можливість сприймати кольоровий спектр який називають CIEXYZ (рис 1.5).

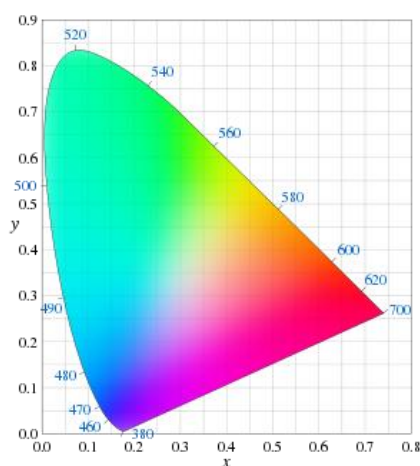


Рисунок 1.5 - Кольоровий спектр CIEXYZ

Основними варіаціями які негативно впливають на зображення

- просторова варіація в освітленості сцени,
- нефіксована геометрія сцени (місце розташування та орієнтація джерела світла, камери та поверхонь у сцені),
- різні джерела освітлення сцени (включаючи комбінацію різних джерела світла в одній сцені)
- використання невідповідних масивів кольорових фільтрів (CFA)

Узагальнену схему створення зображення (рис 1.6) має сталу будову але потрібно розуміти що різні виробники можуть вносити зміни які вони вважають доцільними[20].

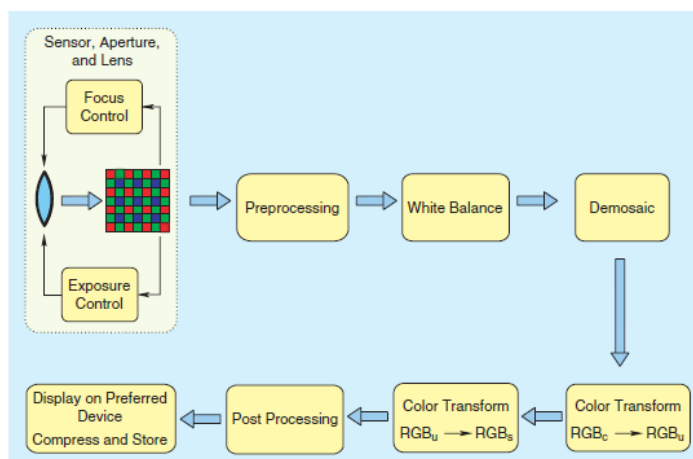


Рисунок 1.6 - Схема створення цифрового зображення

Сучасні сенсори поділяються на дві групи – перша ґрунтується на технології КМОН(комплементарна структура метал-оскид-напівпровідник), і друга на технології ПЗЗ (прилад із зарядним зв'язком)(рис 1.7).



Рисунок 1.7 - Зовнішній вигляд КМОН і ПЗЗ сенсорів

Обидва сенсори працюють по принципу перетворення світла в електричний заряд і подальше перетворення в електричний сигнал. ПЗЗ матриця на виході має аналоговий сигнал для якого потрібно додатково АЦП, КМОН датчик видає зразу цифровий сигнал. Нижче представлена таблиця порівняння основних параметрів матриць ПЗС и КМОП.

Таблиця 1.1 Порівняння основних параметрів матриць ПЗС и КМОП

Параметр	ПЗЗ-матриця	КМОН-матриця
Вихідний сигнал пікселя	Електрони	Напруга
Вихідний сигнал пристрою	Напруга (аналоговий сигнал)	Біти (цифровий сигнал)
Коефіцієнт заповнення	Високого рівня	Середнього рівня
Неузгодженість підсилювачів	Не застосовується	Середнього рівня
Системний шум	Низького рівня	Від помірного до високого рівня
Складність системи	Висока	Низкая
Складність датчика	Низька	Висока
Компоненти камери	Печатна плата, мікросхеми, об'єктив	Мікросхема и об'єктив
Чутливість	Помірного уровня	Поліпшеного уровня
Динамічний діапазон	Високий	Помірний
Однорідність	Висока	Від низької до помірної
Швидкість	Від помірної до високої	Найбільша
Надання неповного зображення (Windowing)	Обмежене	Необмежене
Антиблукінг	Від хорошого до відсутнього	Хороший

Хоча ПЗЗ матриця має більшу чутливість, виробники надають перевагу КМОН сенсорам через простішу будову і більш дешевше виробництво. Також КМОН сенсори через низьке енергоспоживання в статичному режимі і великій швидкості запису (до 500 кадрів\секунду) є більш функціональними. Через наявність у кожному пікселі не тільки світлочутливого елемента а і підсилювача КМОН -сенсор ще називають сенсором активного пікселя. Заряд одразу не переноситься на запам'ятовуючий регістр, кожен елемент адресується окремо від інших. Розглянем більш детально будову КМОН сенсора(рис 1.8).



Рисунок 1.8 - Роль КМОН-датчика в цифровой камері

При відкритому затворі фотони світла збираються на сенсорі, відбувається накопичування фотонів на кожному пікселі датчика - з допомогою лінзи встановленої на кожному пікселі концентруються у фото чутливу зону. При цьому відбувається фільтрація кольору(рис 1.9).

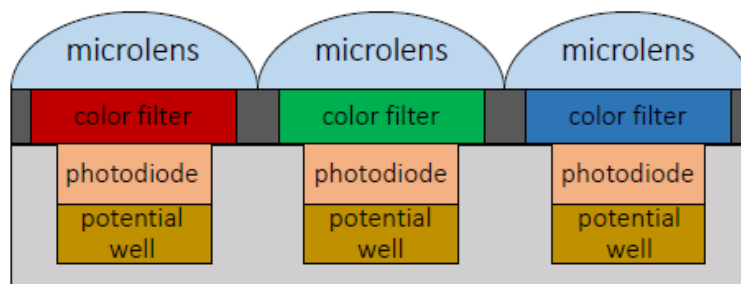


Рисунок 1.9 - Структура пікселя КМОН – сенсора

Світлочутливий елемент накритий фільтром який пропускає світло певної довжини (Червоне, Зелене, Синє, часом Розове). Для створення кольорового простору що сприйнятливий для зору людини кольорові фільтри розташовують у певному порядку. Так утворюється масив кольорових фільтрів (рис 1.10).

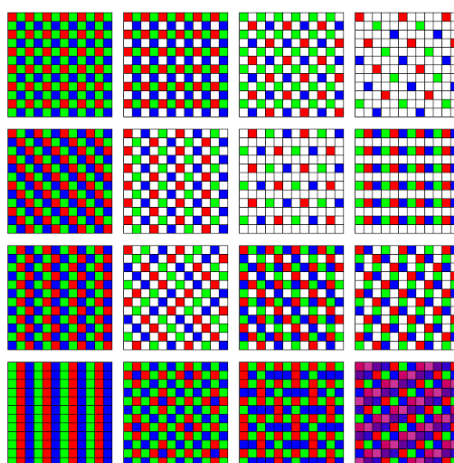


Рисунок 1.10 - Види масивів кольорових фільтрів

В результаті, ігноруються  $2/3$  фотонів на кожному пікселі, але це робить процес більш дешевшим і простішим (потрібно вимірювати 3 кольори одним сенсором). Пропущені значення будуть обчислені під час процесу що називається демозаїка. Це важливий момент який ми розглянемо більше детально. Також додатково може бути використаний інфрачервоний фільтр, який перешкоджає прониканню інфрачервоних хвиль. Фотон

попадаючи на фотодетектор вибиває електрони ,які накопичуються у так званих потенціальних ямах ,що створює різницю потенціалів. По синхронізуючим сигналам генератора імпульсів відбувається одночасне зчитування накопичуваних зарядів, всіх або окремих за вказівкою оператора. Отриманий сигнал буде слабким тому додатково пройде через підсилювач щоб потрапити на вхід АЦП .

Контроль витримки потребує характеристику інтенсивності світла на зображенні .Не достатня або задовга витримка вплине на кольорову гамму вихідного зображення. Залежно від енергії виміряної сенсором система контролю витримки змінює розмір апертури і швидкість затвору. Система фокусу представляється у використанні джерела інфрачервоного світла, розташованого недалеко від лінз, яке б'є променем для оцінки відстані.

Сире зображення отримане із сенсора потребує попередньої обробки перед кольоровою візуалізацією:

- усунення дефективних пікселів ,вони анулюються і значення замінюються інтерполяцією значень пікселів-сусідів.
- Деякі камери потребують щоб данні сигналів були лінійними, з цим справляються оптико-електричні функції конвертори.
- Компенсація темного шуму. Шум що виникає внаслідок тепла генерованого електронами, особливо помітний при малій витримці. Тоді енергія тепла помилково вимірюється сенсором як енергія світла що призводить до спотворень. Усувається оцінкою шуму на темному зображенні і віднімання значення середнього арифметичного шуму від всього зображення. Один із компонентів  $n$  в (1.1).
- Компенсація спалаху. Світло від спалаху відбивається і попадає назад у об'єктив що призводить до засвічувань і плям. Усувається відніманням фіксованого проценту середнього значення енергії сигналу у сусідніх пікселів. Також є компонентом  $n$  в (1.1).



В наслідок багатьох років еволюції людський система зору навчилася визначати колір незалежно від виду джерела світла. Наприклад дивлячись на лист паперу під лампою накаливання і флуоресцентною лампою людина однаково скаже що лист білий. На жаль у камер не має фори у декілька тисячоліть, і це призводить до зсуву у кольоровому спектрі камери . Цей феномен називається сталістю кольорів. Щоб передати справжній колір сцени потрібно мати інформацію про освітлення ,получити її автоматично, що деколи буває неможливо або складно. Зйомка із спалахом дещо полегшує завдання – нам відома інтенсивність і вид освітлення.

Є декілька підходів у настроюванні балансу білого .Перший – метод усереднення . Ще один підхід – прийняття сірого світу. Метод передбачає що сцена зображення в середньому нейтрально сіра .Припущення дотримується коли інтенсивність всіх трьох каналів однакова :  $R = G = B$ . Тоді середнє значення кожного каналу кольору буде відтінком світла зображення.

Найбільш обчислювально-складний крок у процедурі формування цифрового зображення .Полягає у відбудуванні значень пікселів які не були виміряні відтворення повних каналів зображення і з масиву кольорових фільтрів інтерполяцією (рис 1.11).

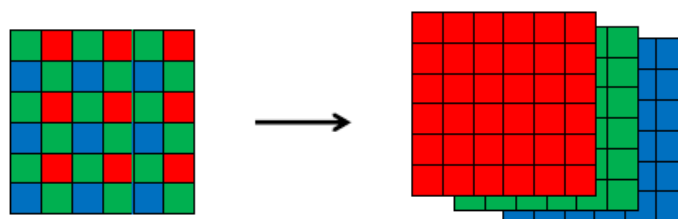


Рисунок 1.11 – Демозайнінг

Більшість алгоритмів демозайнігу становить комерційну таємницю і покриті патентами.

На цьому етапі зображення зберігається у форматі який підходить для зберігання і передачі, а не для відображення. Через те, що спектральна чутливість сенсора відрізняється до кольорових рецепторів в очі людини, зображення потребує трансформації у колориметричний простір CIE XYZ. Цей простір був експериментально сконструйований і описує візуальне сприйняття людини. Для вимірюваного значення пікселя і відповідному колориметричному значенню, має існувати трансформація що співвідносить кольорову характеристику значень виміру камери до простору CIE XYZ.

Зрендерені кольорові простори спроектовані для візуалізації, основна ціль – краса зображення а не детальне відображення сцени (хоча може буде і інакше). Відбувається зміна динамічного діапазону при переході із 12 біт до 8 біт, обмеження гамми  $RGB_C$ .

Наступні кроки це – пост-обробка і зберігання. Пост-обробка плаваюча частина, може включати багато процесів які «наводять красу». Також після демозаїки можуть утворитись артефакти які будуть усуватись, ведеться робота над збільшенню різкості зображення, адже суб'єктивно людина вважає що різке зображення більш краще і якісніше ніж розмите. Зменшується низькочастотна варіація зображення.

Залежно від дисплею на якому має відбуватись відображення потрібно виконувати трансформацію кольорового простору. Високоякісні професійні камери зберігають у власних, вручну-створених форматах по типу TIFF/EP. Формат вміщує інформацію про опто-електронну функцію передачі, освітлення, спектральну чутливість, внутрішні параметри камери і саме зображення у формі мозаїки. У простіших камерах використовують стандарт Exchangeable image file format (офіційно Exif).

					ВМ61.030004.001 ПЗ	Аркуш
Изм.	Лист	№ докум.	Подпись	Дата		23

## 1.6 Доцільність створення системи

Так як шум є невід’ємною частиною і залежить від апаратних і програмних засобів, методи для його усунення без втрати важливих структур і інформації потребують визначення статистичної характеристики, апріорних даних цього шуму ,що є тривалим і складним лабораторним процесом.

Доцільність системи виходить із того що ,на даний момент ,підходи у вирішенні такого типу задач через величезне розмаїття видів сенсорів і ПЗ що беруть участь у зйомці, потребують залучення великої кількості фінансових і людських ресурсів для реалізації конкуренто-спроможного результату. Тому, кінцева ціль – розробка системи відновлення зображень яка б показала хороші результати на реальних зображеннях мала хорошу узагальненість і гнучкість незалежно від джерела зображення(сканер, смартфон, професійна камера, камера відео нагляду, рентген-апарат і інші). Включала можливість додаткового адаптування і калібрування алгоритму відновлення під бажаний вид і архітектуру сенсора пристрою. Також виникла потреба у розробці системи із можливістю до розвитку і покращення результату.

Гонка останніх років призвела до вимоги у покращенні якості зображень у кожній новій моделі, що ,у світі останніх тенденцій - новий апарат-флагман кожні 6-8 місяців, призводить до потреби у швидких, націлених на результат, гнучких підходах актуальність яких буде довготривалою .Також хочу зазначити що дослідження у цій сфері ,хоч і ведуться активно, але в середньому для винаходження або покращення підходу потребується 3-4 роки ,тобто за цей час вийде приблизно 6 поколінь апаратів

## Висновок

У 1 пункті дипломного проекту було розглянуто загальні положення які потрібні для ясного зрозуміння специфіки поставленої задачі. В результаті проведеного аналізу було визначено актуальну модель шуму для задач симуляції, прийнято до уваги особливості проблеми процесу сліпого відновлення і повний цикл створення цифрового зображення.

					ВМ61.030004.001 ПЗ	Аркуш
Изм.	Лист	№ докум.	Подпись	Дата		25

## 2 ОГЛЯД І ПОРІВНЯЛЬНИЙ АНАЛІЗ ІСНУЮЧИХ МЕТОДІВ СЛІПОГО ВІДНОВЛЕННЯ ЗОБРАЖЕНЬ

На даний момент підходи до відновлення і покращення зображень поділяються на такі групи [40]:

1. Локальні методи фільтрації, в тому числі:

- гаусівська модель згладжування [3], де згладжування вимірюється інтегралом Діріхле :  $\int |Du|^2$ .
- фільтр околів Ярославського [4] та Susan фільтр [5];
- білатеральний фільтр [6].

2. Методи, які ґрунтуються на диференціальних рівняннях з частковими похідними (ДРЧП), включаючи:

- модель анізотропної фільтрації [7];
- модель повної варіації Рудіна-Ошера-Фатемі [8];

3. Фільтри в частотній області, що включають:

- локальні адаптивні фільтри в областях перетворення [9];
- жорсткий та м'який трешолдинг [10,11];

4. Алгоритм нелокального усереднення (NL-means) [13].

### 2.1 Локальні методи фільтрації

Шум деякого зображення  $u$  інтерполюється як кусочно-постійна функція в обмеженій області.

#### 2.1.1 Фільтр Гауса

Часто використовується у задачах відновлення і являє собою згортку із використанням заданої функції розмиття точки (лінійним ядром). Формулюється наступним чином :

					ВМ61.030004.001 ПЗ	Аркуш
Изм.	Лист	№ докум.	Подпись	Дата		26

$$G_h(x) = \frac{1}{(4\pi h^2)} \cdot e^{\frac{x^2}{4h^2}} \quad (2.1)$$

Шум зменшується в наслідок згладжування усередненням[40] .Для пікселя із нульовим індексом розмиття фільтром Гауса обчислюється наступним чином :

$$G_h * n(0) = \sum_{i \in I} \int_{P_i} G_h(x) n(x) dx + \sum \epsilon^2 G_h(i) n_i \quad (2.2)$$

де  $n(x)$  інтерпольована кускова функція,  $P_i$  пікселі з центром у  $i$  розміром  $\epsilon^2$ ,  $G_h(i)$  - середнє значення функції  $G_h$  у пікселі  $i$  [1].

### 2.1.2 Фільтри околів

Очищенні значення пікселів отримуються внаслідок зваженого усереднення пікселів сусідів в деякому околі одноканального (сірого) зображення. Окіл рівня сірого :

$$U(i, h) = \{j \in I | u(i) - h < u(j) < u(i) + h\} \quad (2.3)$$

Фільтр локальний тільки у області яскравості. Широко використовується наступний середньозважений фільтр:

$$NH_h u(x) = \frac{1}{C(x)} \int u(y) e^{\frac{|u(y) - u(x)|^2}{h^2}} dy \quad (2.4)$$

де  $\Omega \subset R^2$  є відкритою та обмеженою множиною,  $C(x)$  – коефіцієнтом нормалізації.

### 2.1.3 Білатеральні фільтри

Фільтр, за авторством С. Tomasi та R. Manduchi [6] був представлений у 1998 році [40]. Полягає у використанні двох гаусівських фільтрів: один фільтр працює в просторовій області, а інший в області інтенсивності. Таким чином, не тільки просторова відстань, а й відстань інтенсивності також важлива для визначення вагів. Для даного зображення  $u(x)$  у пікселі  $x$ , вихідний білатеральний фільтр може бути знайдений як:

$$B(x) = \frac{1}{C(x)} \sum_{y \in N(x)} e^{-\frac{\|y-x\|^2}{2\sigma_d^2}} e^{-\frac{|u(y)-u(x)|^2}{2\sigma_r^2}} u(y) \quad (2.5)$$

де  $\sigma_d$  та  $\sigma_r$  параметри контролю зменшення вагів у просторовій (відстань) області та області інтенсивності (радіометрична),  $N(x)$  – просторовий окіл пікселя  $u(x)$ , та коефіцієнт нормалізації  $C(x)$ . Фільтр часто використовується, але є проблемним – не існує чіткого підходу у оприділенні розміру вікна околу усереднення, тільки шляхом експериментів. При занадто малому розмірі вікна відновлення обмежене, а при занадто великому пікселі мало зв'язані і алгоритм перестає бути ефективним [6].

## 2.2 Методи на основі диференціальних рівнянь з частковими похідними (PDE)

### 2.2.1 Анізотропне дифузійне рівняння

Алгоритми часткових диференціальних рівнянь усувають шум зображення із збереженням важливих країв(структур)[40]. Виконується шляхом дифузійного процесу, в якому зображення багаторазово фільтрується і локально накладається до основного сигналу зображення. Нехай  $I(x, t)$  – зображення в дифузійному процесі за часом  $t$ ,  $I(x, u): R^2 \rightarrow R^+$  – початкове зображення в безперервній області,  $x \in R^2$  – положення пікселя в зображенні 2D. Раніше, ізотропне дифузійне рівняння :

$$\frac{\partial I(x, t)}{\partial t} = \operatorname{div}(\nabla I), \quad I(x, \infty) = u \quad (2.6)$$

використовувалось для знешумлення зображення. Доведено, що зміна зображення згідно з цим анізотропним дифузійним рівнянням еквівалентна до фільтрування зображення фільтром Гаусса [7].

### 2.2.2 Повна варіація

Представлена Rudin, Osher та Fatemi [8]. Розглядає проблему мінімізації :

$$E(u) = \lambda \|u - v\|_{L^2(\Omega)}^2 + R(u) \quad (2.7)$$



де  $v(x) = u(x) + n(x)$ ,  $\lambda > 0$ ,  $\Omega$  – область на якій  $v$  визначається, та  $R(u)$  – оператор регуляризації. Раніше зусилля були сфокусовані на операторі  $R(u)$ , який ґрунтувався на найменших квадратах. Оператор регуляризації підвищує сходимість функції і нівелює уривчастість.  $BV(\Omega)$ , простір функції, що інтегрується з обмеженою повною варіацією  $TV_{\Omega}(u) = \int |\nabla u|$ . При заданому спотвореному знімку  $v(x)$ , було запропоновано відновити початкове зображення  $u(x)$  рішенням обмеженої проблеми мінімізації:

$$\arg \min TV_{\Omega}(u) \quad (2.8)$$

при умові обмеження шуму:

$$\int_{\Omega} (u(x) - v(x)) dx = 0 \text{ та } \int_{\Omega} |u(x) - v(x)|^2 dx = \sigma^2 \quad (2.9)$$

Рішення  $u$  має бути настільки систематичним (правильним) як це можливо у сенсі повної варіації, в той час як різниця  $u - v$  розглядається (тракується) як помилка. Обмеження задає правильне середнє значення та дисперсію для  $u - v$ , але не гарантує, що воно буде схоже на шум [10].

Отже, маємо:

$$\arg \min (TV_{\Omega}(u) + \lambda \int_{\Omega} |v(x) - u(x)|^2 dx) \quad (2.10)$$

де  $\lambda$  коефіцієнт Лагранжа. Наведена функція строго опукла і напівбезперервна знизу. Тому існує унікальний мінімум який можливо обчислювати. Параметр  $\lambda$  контролює компроміс між правильністю та точністю. Метод повної варіації показав хороші результати у відновленні

високочастотної інформації із дефектних зображень, а саме відтворення ребер об'єктів і їх кутів[40].

### 2.3 Фільтри в частотній області

Спостережуване зображення  $v$  можна записати:

$$v(i) = u(i) + n(i) \quad (2.11)$$

де  $u$  – оригінальне зображення визначене на  $R^n$ ,  $n(i)$  – незалежний і однаково розподілений з нульовим середнім значенням і сталою дисперсією  $\sigma^2$ . Нехай  $B = \{g_\alpha\}_{\alpha \in A}$  буде ортогональним базисом  $R^n$ . Визначимо

$$v_B(\alpha) = \langle v, g_\alpha \rangle, \quad u_B(\alpha) = \langle u, g_\alpha \rangle, \quad n_B(\alpha) = \langle n, g_\alpha \rangle$$

як скалярні добутки. Тоді

$$v_B(\alpha) = u_B(\alpha) + n_B(\alpha) \quad (2.12)$$

трансформований шумовий процес. Шумові коефіцієнти  $n_B(\alpha)$  залишаються некорельованими та дорівнюють нульовим середнім значенням, але дисперсії помножуються на  $\|g_\alpha\|^2$ . Частотні діапазонні фільтри застосовуються незалежно до кожного трансформованого коефіцієнту. Під час відновлювання зображення, просто виконується зворотне перетворення нових коефіцієнтів. Шумові коефіцієнти  $v_B(\alpha)$  перетворюються на  $a(\alpha)v_B(\alpha)$ , де  $a(\alpha)$  часто обмежуються значенням 0 або 1. Це нелінійний алгоритм через те, що  $a(\alpha)$  залежить від значення  $v_B(\alpha)$ [40]. Зворотне перетворення дає оцінку

$$\hat{U} = DV = \sum_{a \in A}^a (a) v_B(a) g_a \quad (2.13)$$

Частотний діапазонний фільтр використовує базис Фур'є. Задача знаходження ідеального базису для кожного конкретного прикладу залишається проблемою в обробці зображень [9].

### 2.3.1. Гранична вейвлет обробка

Метод розроблений Д.Донохо [10]. Нехай  $B = \{g_a\}_{a \in A}$  ортогональний вейвлет базис [11]. Для визначення  $a(a)$  вибирається певний поріг  $\mu$ , тобто:

$$a(a) = \begin{cases} 1, & |v_B(a)| > \mu \\ 0, & |v_B(a)| \leq \mu \end{cases} \quad (2.12)$$

Метод базується на принципі, що зображення мають великі вейвлет коефіцієнти, які зберігаються, в той час як шум описується малими коефіцієнтами, якими можливо знехтувати[40]. Пропускна здатність апроксимації  $u$  впливає на ефективність методу. Донохо [11] показав, що ці ефекти можуть бути частково зменшені, якщо використовується м'яка гранична обробка  $SWT_{\mu}(v)$ . Відбувається зберігання структур вейвлетів і зменшення коливання околів ,якщо оператор м'якої граничної обробки непереривний.

## 2.4 Нелокальні методи

Особливість методів полягає у відсутності локальних обмежень. Розроблені і представлені у роботах М. Махмоуді та Д. Сапіро [17]. На основі матриць подібності різних регіонів зображення відбувається знешумлення.

### 2.4.1 Алгоритм нелокального усереднення

Нехай  $v(i)$  та  $u(i)$  будуть досліджуваними зображеннями з шумом та чистим зображенням, відповідно, де  $i$  – індекс пікселя. Відновлені значення можуть бути отримані як середньозважена величина усіх сірих значень в зображенні (індексовані у множині  $I$ ):

$$NL(v)(i) = \sum_{j \in I} \omega(i, j) v(j) \quad (2.13)$$

де  $NL(v)(i)$  – відновлені значення в пікселі  $i$ . Ваги відображають суму подібності між околами кожної пари пікселів, що беруть участь у розрахунку  $i$  та  $j$ :

$$\omega(i, j) = \frac{1}{Z(i)} e^{-\frac{\|v(N_i) - v(N_j)\|^2}{h^2}} \quad (2.14)$$

де  $\|\cdot\|$  – описує зважену відстань Гауса і  $Z(i)$  – нормуючий множник  $Z(i) = \sum \omega(i, j)$ . У наведеному вище рівнянні  $v(N_i)$  – вектор значень околу пікселів  $v(N_i) := (v(j))$ ,  $j \in N_i$ , де  $N_i$  визначає окіл пікселя  $i$  як стандартний квадратний блок зумовленого розміру навколо  $i$ .  $h$  – глобальний згладжувальний параметр, який контролює кількість розмивання, введенного в

процес шумозаглушення. Для більш високих значень шуму, присутнього в зображенні, встановлюється більше значення  $h$ . Подібність між двома пікселями  $i$  та  $j$  залежить від подібності інтенсивностей сірого рівня векторів  $v(N_i)$  та  $v(N_j)$ . Пікселі з аналогічним оком сірого рівня до  $v(N_i)$  в середньому матимуть більшу вагу. Для обчислення схожості між двома блоками використовується гаусівськи зважена Евклідова відстань.

## Висновок

В результаті аналізу було виявлено що більшість методів відновлення зображень зосереджені на півтонових зображеннях з додаванням штучного шуму[40] (рис. 2.1) .

Значення PSNR для різних методів			
Метод	Середнє значення PSNR		
	Без обробки	Зображення зі штучним шумом	Зображення з реальним шумом
Гаусівська модель згладжування [3]	24,54	25,45	24,57
Фільтр околів Ярославського [4]	24,54	26,02	24,83
Susan фільтр [5]	24,54	26,04	25,00
Білатеральний фільтр [6]	24,54	27,12	25,06
Метод анізотропної фільтрації [7]	24,54	26,67	25,02
Метод повної варіації Рудіна-Ошера-Фатемі [8]	24,54	27,03	25,12
Локальні адаптивні фільтри в областях перетворення [9]	24,54	27,65	25,17
Жорсткий трешолдинг [10]	24,54	26,08	25,01
М'який трешолдинг [11]	24,54	26,35	25,02
Zhou-Wang вейвлет повна варіація [12]	24,54	27,89	25,41
Алгоритм нелокального усереднення (NL-means) [13]	24,54	27,56	25,39

Рисунок 2.1 – Порівняння продуктивності методів відновлення

І лише мала частина призначається для природних кольорових фотографій, отриманих цифровою камерою з реальним шумом. Але ці методи є або надто складними, або не можуть вирішити проблему зменшення шуму в достатній мірі. Тому необхідно продовжити дослідження в напрямку визначення чинників, які впливають на зменшення шуму та підвищення якості зображень з реальним шумом.

## 3 ПОНЯТТЯ НЕЙРОННОЇ МЕРЕЖІ

### 3.1 Штучна нейронна мережа

Штучна нейронна мережа (ШНМ, англ. artificial neural network ANN), за визначенням – це обчислювальна система, яка використовує мережу функцій для розуміння та переведення даних однієї форми в потрібний вихід, як правило іншої форми. Особливістю ШНМ є можливість навчатись і покращувати свою продуктивність відносно часу і даних. ШНМ можуть розглядатися як спрямований граф зі зваженими зв'язками, у якому штучні нейрони є вузлами. Нейронна мережа працює за принципом який є схожий до навчання дітей: вона сприймає набір прикладів-завдань і відповідей до них (наприклад :набір зображень фруктів і масив їх назв ).На основі наданих прикладів вона створює функцію зв'язку між певним прикладом і його значенням, запам'ятовує ключові особливості. Нейронні мережі мають здібність до узагальнення ,тобто трансформують об'єкт до форми яка є узагальненням форм даних які були використанні під час навчання . Звичайні штучні нейронні мережі (рис. 3.1) погано масштабуються у випадку з зображеннями великих розмірів.

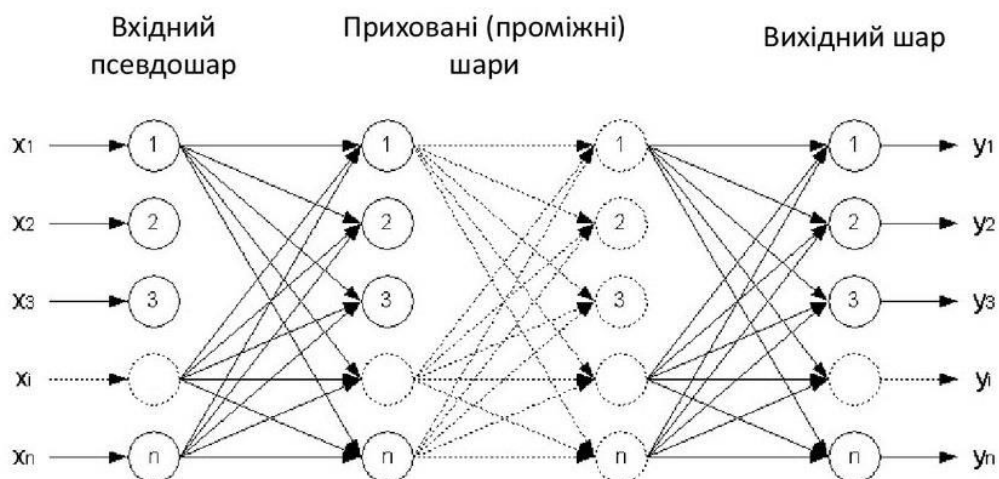


Рисунок 3.1 - Багатошарова нейронна мережа

Так, в системі комп'ютерного зору CIFAR-10, картинки становить  $[32 \times 32 \times 3]$  (32 - ширина, 32 - висота, 3 – канали кольорів), тому один повністю підключений нейрон в першому прихованому шарі звичайної нейронної мережі має вагу  $32 * 32 * 3 = 3\,072$ . Повнозв'язна структура не масштабується для великих зображень. Наприклад, при розмірі картинки  $[200 \times 200 \times 3]$  кількість зв'язків повністю підключеного нейрона буде дорівнювати 120 000. Мережі складаються із багатьох сотень нейронів, тому такий підхід призведе до величезної кількості параметрів які важко навчати[41].

### 3.2 Згорткові нейронні мережі

Для обробки складних даних (наприклад: зображення) були створені згорткові нейронні мережі (рис. 3.2). Їхня архітектура була адаптована під певні задачі для економії пам'яті і покращення продуктивності мереж.

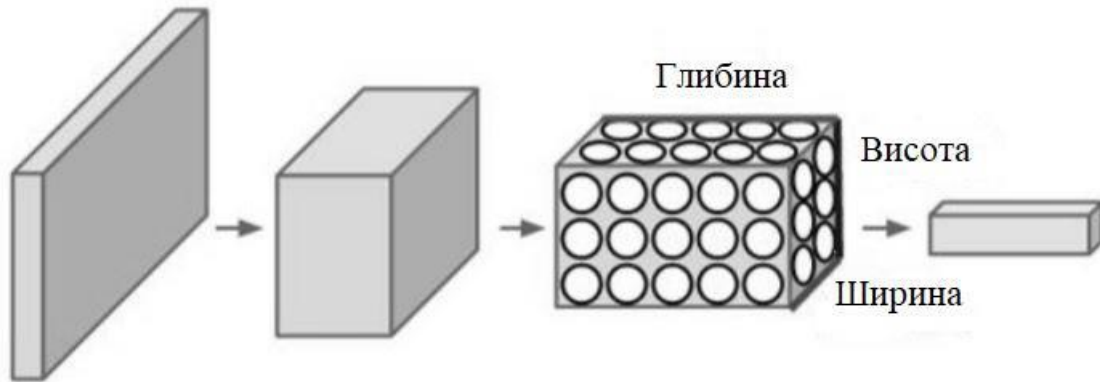


Рисунок 3.2 – Згорткова нейронна мережа

Нейрони у шарах ЗНМ розташовуються у трьох вимірах :ширині ,висоті, глибині, і створюють тензор. Поданий на вхід сигнал підлягає багаторазовій трансформації для стиснення і узагальнення його ознак. Вихід мережі формується відносно поставленої задачі .Це може бути як і набір класів, неперервне число чи зображення.

Основною операцією ЗНМ вважається шар згортки. Використовується набори лінійних ядер (матриць) що виконують функцію фільтру. Для кожного фільтру задається його розміри(ширина і висота), зазвичай вони не великі ([3x3], [5x5], [7x7]). Згортка відбувається ковзанням лінійного ядра по зображенню (попередньому шару) і обчисленням скалярного добутку сигналу і матриці. У міру проходження фільтра по ширині і висоті зображення, складається двомірну активаційна карта, яка надає відгук цього фільтра на кожній просторовій позиції (рис. 3.3).



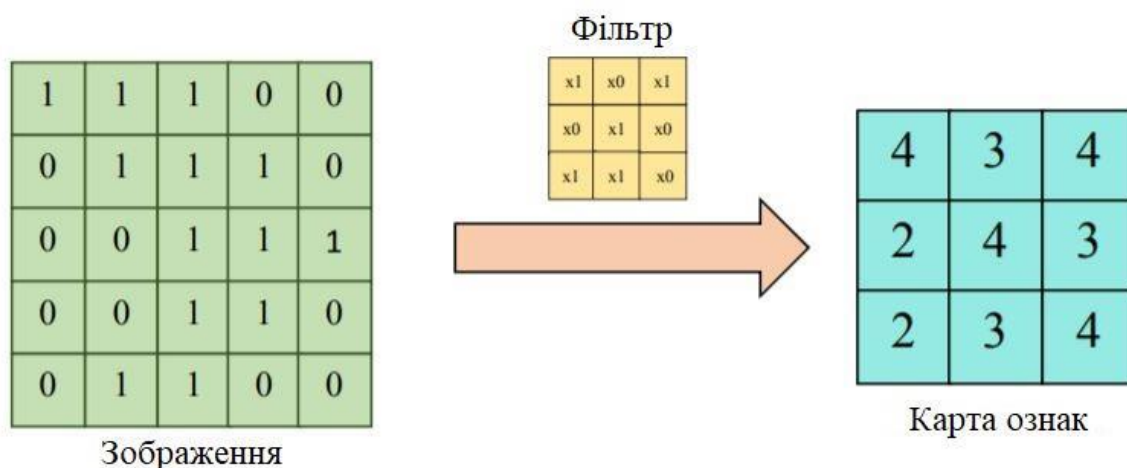


Рисунок 3.3 – Формування карти ознак .Операція згортки

Фільтри згортки узагальнюють візуальні особливості і структури на зображенні, такі як ребра, спрямованості, плямистість .Кожне лінійне ядро формує свою двомірну активаційну карту ознак. Ці карти будуть формувати вихідний об'єм.

### 3.3 Навчання нейронної мережі

Ключовою особливістю мозку людини є здатність до навчання і розвитку. У контексті нейронної мережі навчанням можна назвати процес поступового налаштування ваг і зв'язків нейронів ,що складають архітектуру і виконують поставлену задачу із деякою мірою якості .Зазвичай налаштування(такі як вага кожного нейрону) відбувається в наслідок представлення мережі навчальних прикладів .Властивість мереж навчатися за поданими прикладами робить їх більш досконалими у порівнянні з системами, що працюють за прописаними заздалегідь правилами. Серед усіх чинних методів навчання мереж можна виділити два класи: детермінований та стохастичний. Детермінований клас методів інтерактивно корегує параметри мережі, спираються на стан поточних параметрів, величини

входів, фактичних та бажаних виходів. Клас методів у якому параметри мережі змінюються випадково і залишаються лише позитивні зміни називається стохастичним[41].

Одним із найпопулярніших підходів до навчання нейронних мереж є навчання із учителем: задача машинного навчання у реалізації функції виведення на підставі відомих навчальних даних.

Тренувальний набір складається із пар об'єктів : прикладу (зазвичай вектора) і бажаного результату (контрольний сигнал).Обраховується значення функції до розгляду прикладу порівнюються із значенням очікуваного виходу, обраховується функція втрат і відносно неї корегуються ваги нейронів мережі

Алгоритм навчання з учителем :

1. Набір даних розділений на підвибірки : навчальну і тестову у відношенні 70/30.
2. Обчислити функцію похибки на навчальній вибірці.
3. Корегування ваг .
4. Повторювати другий і третій пункт поки значення похибки не досягне бажаного рівня .
5. Провести контрольну оцінку продуктивності мережі на тестових даних із якими мережа не знайома.

Для хорошої продуктивності потрібна велика навчальна вибірка :для кращого узагальнення і гнучкості.

На Рисунку 3.4 зображено два криві: криву значень помилок на навчальному наборі (величина якої задана горизонтально) та криву значень помилок на тестовому або перевірочному наборі (яка має фіксований розмір). Чим більше даних використовується для навчання, тим більше помилок

видасть архітектура мережі, що відповідатимуть навчальним даним. У той же час навчальні дані стають більш схожими на справжній розподіл даних, які слід зафіксувати за навчальними даними. У певний момент часу помилка на навчальному та тестовому наборах повинна бути приблизно однаковою.

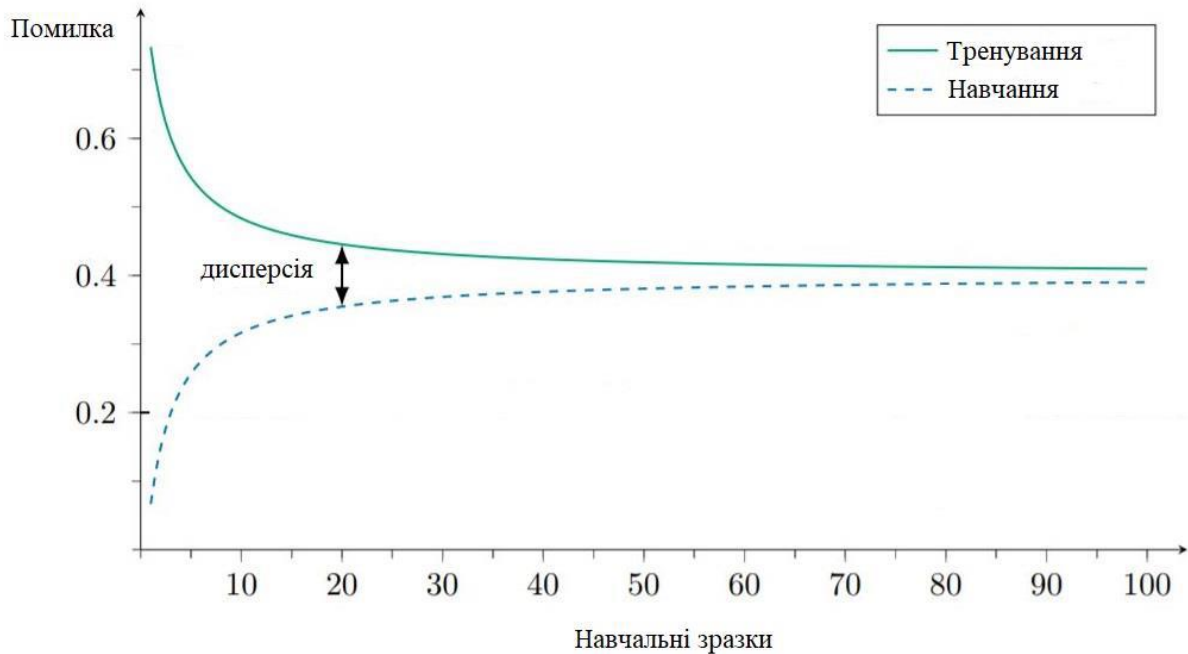


Рисунок 3.4 – Процес навчання мережі

Крива процесу навчання для встановлення є індикатором правильності роботи мережі [41], якщо велика кількість навчальних даних без будь-яких змін, що в свою чергу покращує ефективність мережі. Побудувавши криву навчального набору, можна оцінити, чи здатна архітектурна модель мережі відповідати даним для потрібної класифікації помилки. Значення помилки при тестовому наборі ніколи не повинне бути значно нижчим, ніж значення помилки навчального набору. Якщо помилка на навчальному наборі занадто висока, то збільшення даних не допоможе вирішити проблему. Натомість стане зрозуміло, що сама архітектура моделі або алгоритм її навчання потребують коригування. У випадку, коли на графіку крива навчального набору значно перевищує криву тестового (перевірочного) набору даних

необхідно або зменшити роздільну здатність зображень, або ввести додаткові навчальні зразки даних, або збільшити регуляризацию для вирішення даної проблеми.

### 3.4 Функція втрат і оцінка точності

Функція втрат (англ. loss function), також називається функцією помилки або функцією вартості - це функція, яка вираховує дійсне значення для складної події, як-от передбачення приналежності до класу вхідного вектора[41]. Вона використовується для визначення цільової функції. У пролематиці класифікації функція втрат подається як перехресна ентропія з  $L1$  або  $L2$  регуляризациєю. С точки зору теорії інформації, навчанням є мінімізація значень перехресної ентропії стосовно реальних класів і гіпотезами моделі. Значенні втрати даних є позитивним, коли класифікація є неправильною, але значення втрат є числом більшим для більш складних моделей мережі. Якщо дві моделі мереж розпізнають однаковий набір даних, то кращою визнається модель архітектура якої є простішою. Приводом для зображення саме функції втрат методом кривої на графіку замість інших показників якості є те, що тоді графік міститиме більше інформації про якість мережі. Головним недоліком функції втрати є те, що вона не має верхньої межі, як точність, і її важко інтерпретувати. Функція втрати лише показує відносний прогрес навчання, тоді як точність показує абсолютний прогрес[41].

На практиці важливими є також і інші критерії якості:

- швидкість оцінки та аналізу нових зображень, що надаються мережі;
- затримка часу навчання;
- ресурсозатратність;

- стійкість до (не) випадкових відхилень у навчальних даних;
- розмірність архітектури мережі.

## Висновок

Проведений аналіз обчислювальних систем типу нейронна мережа. Взято до уваги основні принципи роботи системи, види мереж, навчання, функції похибок і архітектури.

					ВМ61.030004.001 ПЗ	Аркуш
Изм.	Лист	№ докум.	Подпись	Дата		42

## 4 РОЗРОБКА СИСТЕМИ СЛІПОГО ВІДНОВЛЕННЯ ЗОБРАЖЕННЯ ІЗ ДОПОМОГОЮ НЕЙРОННИХ МЕРЕЖ

### 4.1 Підготовка даних для калібрування системи відновлення

Для навчання нейронної мережі регресувати від зашумленого зображення до чистого потрібен набір даних достатнього розміру який також буде прямо впливати і на якість.

Було висунуто такі вимоги до наборів даних:

- Набір повинен складатись із оригінальних дефектних зображень створених у реалістичних умовах на камерах комерційного виробництва.
- Бажано щоб зображення були зроблені на різних пристроях для кращого узагальнення.
- Також у набір має входити зображення еталони із мінімальною кількістю шуму .Тобто набір даних має складатись із пар - зашумлене зображення і зображення еталон.

Проаналізувавши існуючі системи відновлення було виявлено низьку якість відновлення реальних знімків у випадку використання Гаусовської моделі шуму. На даний момент існують такі набори даних ,що відповідають вимогам : RENOIR[37] і DND[36]. Датасет RENOIR складається із 120 пар зображень зроблених на пристроях Canon T3i ,Canon S90,Xiaomi Mi3.Зображення еталони були отримані внаслідок ручної ,акуратної обробки. Як правило у лабораторних умовах відбувається зйомка сцени із спеціально спланованим освітленням, витримкою і іншими параметрами. Робиться фотознімок сцени ,ведеться робота над максимізацією відношення сигнал-шум. У подальшому зроблений знімок рахується ідеалом і використовується у системах відновлення як еталон.Набір DND представляє собою набір із 50

					ВМ61.030004.001 ПЗ	Аркуш
Изм.	Лист	№ докум.	Подпись	Дата		43

зображень у сирому форматі зроблених на професіональних камерах із різними розмірами сенсорів. Цей набір зазвичай використовується для оцінки якості роботи систем відновлення зображень – еталонні зображення недоступні і не відомі. Для оцінки роботи відновленні зображення відправляються на сайт ,який обчислює якість у метриках PSNR/SSIM.Також взято до уваги можливість створення власного набору із використанням камери смартфона. У сучасних сенсорах мобільних телефонів використовують передові розробки і підходи , по суті зображення формуються ідентичним образом як і стаціонарні камери .Зробивши велику кількість фотографій однієї сцени за ідеальне зображення можна рахувати їхнє середнє арифметичне.

Виникла проблема, для навчання нейронної мережі наявної кількості даних не достатньо - як приклад нейрона мережна U-net[33] була навчена на 1 мільйоні зображень із використанням аугментацій. Незважаючи на комерційну важливість галузі існує невелика кількість наборів. Це викликано складністю і великою вартістю ,що потребується для їх створення . У зв'язку із великою кількістю видів і виробників сенсорів камер неможливо створити набір , який підійде для калібрування моделі під усі специфікації сенсорів .

Для виконання поставленої задачі потрібно збільшити набір даних із дотриманням вказаних вище вимог. Першим варіантом рішення є аугметнація наявних зображень. Аугментація - це процес штучного збільшення кількості даних внаслідок їх перетворень. На прикладі зображень це можна сприйняти як нарізка одного зображення на декілька ,повороти відносно центру ,віддзеркалювання, зсув. Це дозволить суттєво збільшити об'єм вибірки. Приклад аугментації представлений нижче (рис 4.1).

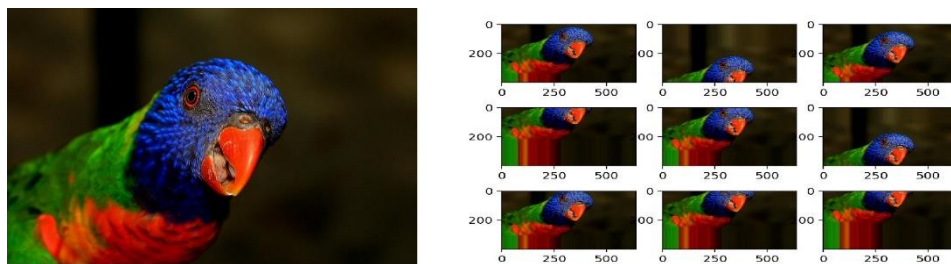


Рисунок 4.1 – Аугментація зображення

Провівши аналіз ,було виявлено недолік процесу аугментації. Незважаючи на збільшення об'єму вибірки її різноманіття і репрезентативність відносно дефектів не змінилась. А це значить що модель лише навчиться узагальнювати наявні шуми і не зможе ефективно працювати на незнайомих зображеннях. Підхід можна використовувати як допоміжний ,але не підходить для повного вирішення задачі.

Взявши до уваги напрацювання у напрямку моделювання шумів які викладені у пункті 1.4 було розглянуто можливість моделювання і накладання шуму який зміг би апроксимувати особливості реального шуму. Фінальне зображення яке спостерігається на дисплеї є наслідком тривалого і складного процесу який присутній у абсолютній більшості сучасних камер. Процес можна розділити на два основних стани, зображення до внутрішньої обробки (так зване сире зображення) і після (демозаїка, баланс білого і інші.). Природа шуму у першому стані характеризується фотонним шумом і електричним шумом , є лінійною залежністю , достатньо зрозумілою і піддається моделюванню. На оборот, другий стан ,згідно із 1.5 ,включає велику кількість трансформацій і є підставою виникнення комплексних і не лінійних спотворень на зображеннях.

Це вказує на доцільність моделювання шуму на зображеннях формату RAW.Фотонний шум, (1.4), є основною формою невизначеності, пов'язаної з вимірюванням світла, властивою квантовій природі світла та незалежності детектування фотонів. Очікувана його величина залежить від сигналу і



становить домінуюче джерело шуму зображення, за винятком умов слабкої освітленості.

Датчики зображення вимірюють опромінення сцени, підраховуючи кількість дискретних фотонів, що падають на датчик протягом заданого часового інтервалу. У цифрових датчиках фотоелектричний ефект використовується для перетворення фотонів в електрони, тоді як плівкові датчики покладаються на світлочутливі хімічні реакції. В обох випадках незалежність випадкових індивідуальних надходжень фотонів призводить до фотонного шуму, залежної від сигналу форми невизначеності, яка є властивістю самого сигналу. У комп'ютерному зорі поширеним наближенням є моделювання шуму зображення як незалежного від сигналу, часто з використанням нульової середньої добавки Гауса. Хоча цієї простої моделі достатньо для деяких застосувань, вона фізично нереальна. У реальних системах візуалізації фотонний шум та інші джерела шуму на основі датчиків вносять різну пропорцію при різних рівнях сигналу, що призводить до шуму, який залежить від яскравості сцени. Індивідуальні виявлення фотонів можна трактувати як незалежні події, що слідує за випадковим часовим розподілом. В результаті підрахунок фотонів є класичним Пуассоновим процесом, а кількість фотонів  $N$ , виміряне даним сенсорним елементом за часовий інтервал  $t$ , описується дискретним розподілом ймовірностей (4.1)[23].

$$P(N = k) = \frac{e^{-\lambda t} (\lambda t)^k}{k!} \quad (4.1)$$

де  $\lambda$  - очікувана кількість фотонів на одиницю часового інтервалу, пропорційне опроміненню на сцені інциденту. Це стандартний розподіл Пуассона з параметром швидкості  $\lambda t$ , що відповідає очікуваному кількості

					ВМ61.030004.001 ПЗ	Аркуш
Изм.	Лист	№ докум.	Подпись	Дата		46

фотонів інциденту. Невизначеність, описана цим розподілом, відома як фотонний шум.

Це наближення, як правило, дуже точне. При малих підрахунках фотонів у фоновому шумі, як правило, переважають інші джерела шуму, незалежні від сигналу, а для більших підрахунків центральна гранична теорема забезпечує, що розподіл Пуассона наближається до розподілу Гауса. Фотонний шум властивий вимірюванню світла, не має параметрів для калібрування і не залежить від інших джерел шуму. В результаті вплив фотонного шуму на зображення можна охарактеризувати за допомогою функції передачі модуляції (оптична передаточна функція), яка пов'язує кількість фотонів та очікувану інтенсивність пікселів. Наведемо приклади оптичної передаточної функції деяких комерційних цифрових камер і сенсорів (рис 4.2).

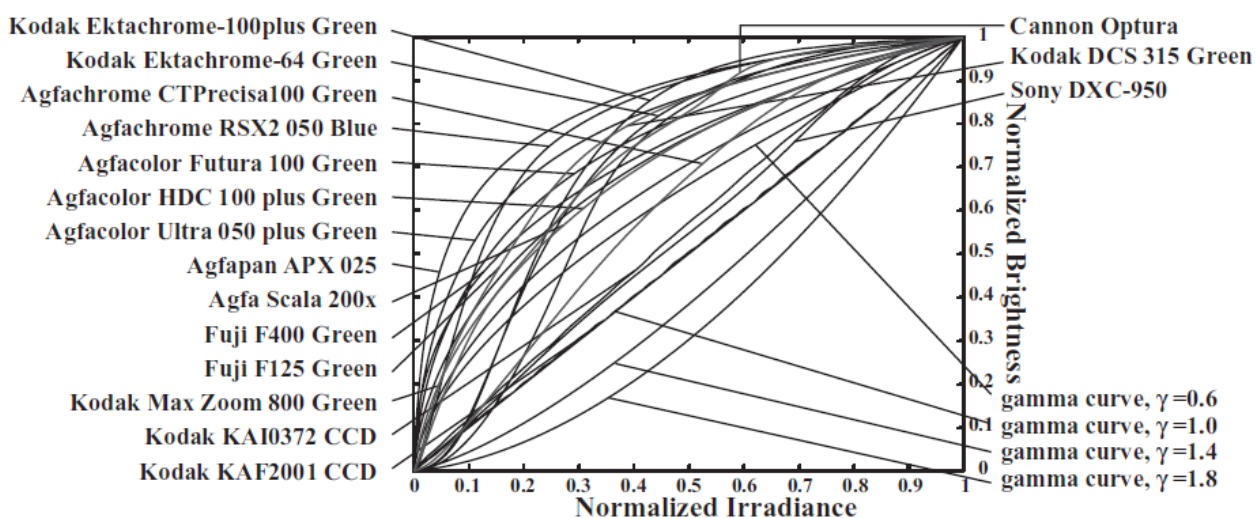


Рисунок 4.2 – Приклади функції передачі модуляції із набору 201 реальних функцій DoRF [39]

Електронний шум, або шум зчитування генерується розташованим на кристалі світлочутливого сенсора підсилювачем перетворення заряду, що зберігається в кожному фотодіоді (пікселі) в аналогову напругу, яке, потім, надходить на АЦП. Шум зчитування можна розглядати, як неминучу «плату»

за зчитування збереженого заряду. Завдяки вдосконаленню конструкцій матриць ПЗС, а також методів синхронізації і вибірки, за останні кілька років ця величина неухильно знижується до 5-10 електронів / піксель.

Шум зчитування пропорційний швидкості зчитування. Плата за швидкість - зростання рівня шуму і, як наслідок, збільшення невизначеності при вимірюванні напруги та меншу кількість розрядів дозволу. Саме тому у камер з повільною розгорткою, в загальному випадку, рівень шуму зчитування менше, а кількість значущих розрядів більше, ніж у швидкодіючих фотоприймачів. Електронний шум не залежить від сигналу і моделюється розподілом Гауса.

Проведемо симуляцію процесу зйомки сцени камерою для моделювання спотворення внаслідок фотонного шуму, електричного шуму із допомогою мови програмування Python версії 3.7.4 . Для простоти почнемо з припущення, що у нас є камера з  $256 \times 256$  пікселями і середнім потоком падаючого світла по 500 фотонів на кожен піксель (рис. 4.3).

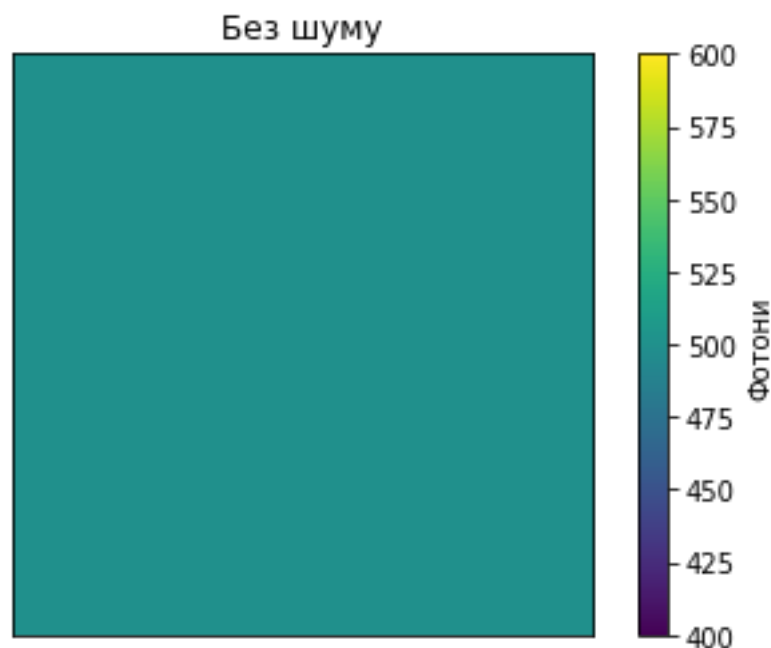


Рисунок 4.3 – Тестове зображення

Шум виникає через те, що кількість фотонів, що падають, не є детермінованим, а скоріше випадковим процесом, статистика якого є пуассонівська. Із допомогою бібліотеки NumPy симулюємо і порівнюємо із початковим зображенням (рис 4.4).

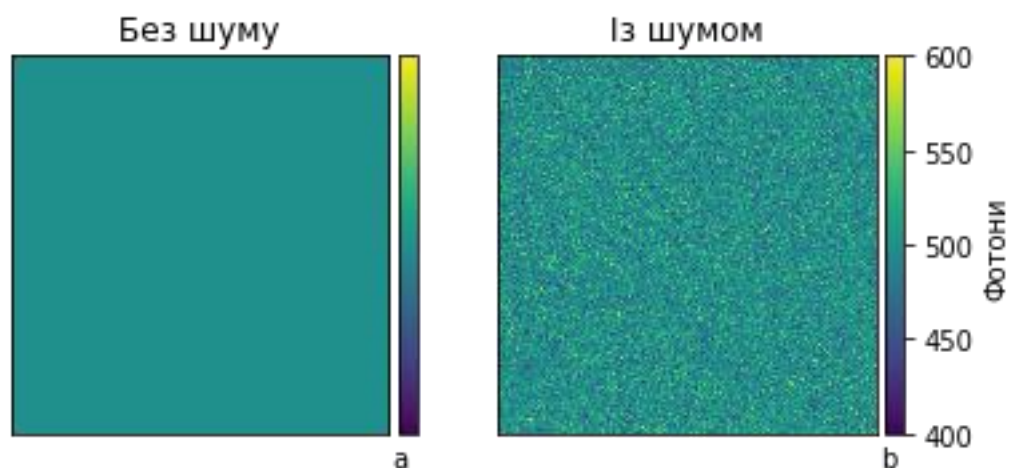


Рисунок 4.4 – Порівняння зображень

Також будуємо графік розподілу інтенсивності фотонів на піксель (рис 4.5).

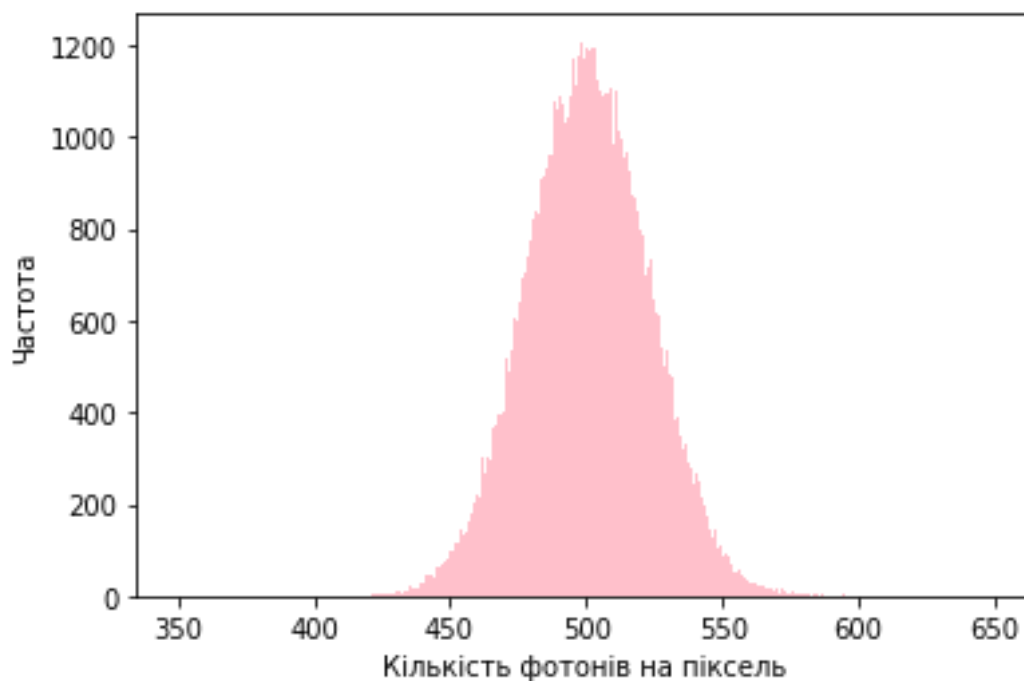


Рисунок 4.5 – Розподіл інтенсивності

Для подальшої симуляції фотоелектронів використовуємо для прикладу ключові характеристики сенсора Sony IMX264 архітектури КМОН включно із функцією передачі модуляції, а саме :

Таблиця 4.1 - Характеристики сенсора Sony IMX264

Вибіркові характеристики сенсора Sony IMX264	Значення
Квантова ефективність	0,69
Чутливість	5.88 ADU/ e-
Розмір пікселю	3,45 мкм
АЦП	12 біт
Показник початкового електричного шуму	2.29 e-

Якщо припустити, що наше світло має довжину хвилі 525 нм, то кількість фотоелектронів є добутком квантової ефективності з реалізацією поля. Результат моделювання падаючих на сенсор фотонів і їх перетворення у заряд електронів представлено на Рисунку 4.6 .

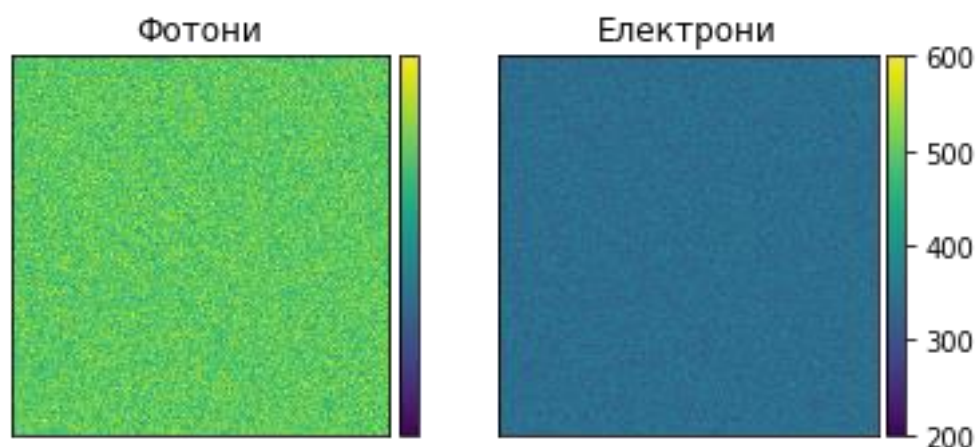


Рисунок 4.6 – Зліва, симуляція потоку фотонів ,справа вихідні фотоелектрони

Тепер змодельюємо електронний шум розподілом Гауса взявши за його стандартне відхилення параметр електричного шуму який вказаний у Таблиці 4.1 .Імітуємо процес зчитування накопленого заряду пікселя накладанням Гаусовського шуму (рис 4.7).

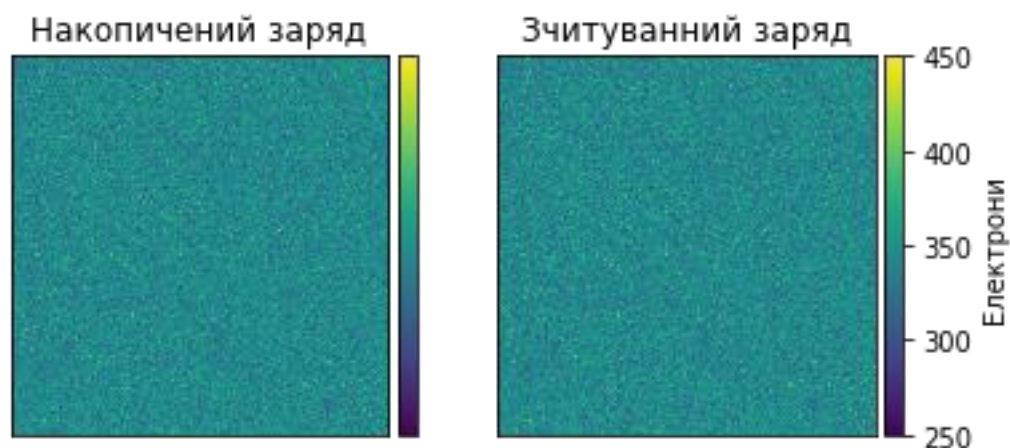


Рисунок 4.7 – Симуляція електричного шуму

Віднімемо зображення після впливу електричного шуму від зображення накопиченого заряду(рис 4.8).

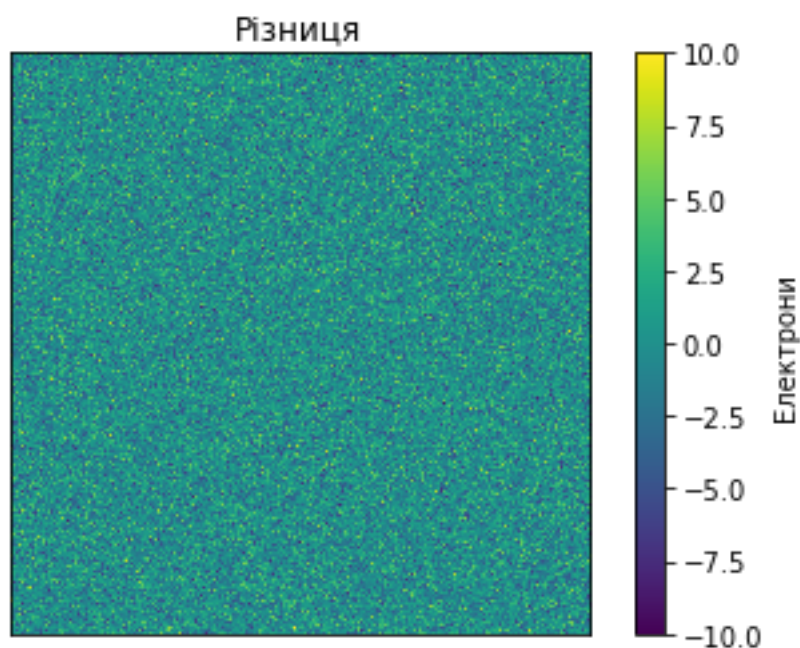


Рисунок 4.8 – Різниця зображень до накладання електронного шуму і після

На даному етапі ми оприділились із реалізацією моделі шуму яка буде використовуватись. Так як реальне зображення в загальному проходить цикл трансформацій які ще більше збільшують складність шуму і робить його просторово і хроматично залежним. Тому ,доцільно змоделювати основні етапи обробки для створення реалістичних штучних спотворених зображень на яких зможе навчатись нейронна мережа. Присутні великі набори зображень які використовуються для задач класифікації і сегментизації. Як приклад набір даних MIRFLICKR[35], що складається із 25000 зображень зібраних із відкритих джерел. Точне походження ,вид сенсора і інші данні про специфіку зйомки не відомі .Зображення «готові» , тобто є вихідним результатом обробки пристрою що їх створив. Це значить що ніякої корисної інформації вони не несуть, але було вирішено використати їх як аналог еталонних зображень до накладання на них моделі шуму нашої реалізації.

Як було сказано раніше ,доцільно використовувати зображення у форматі RAW,тому розробимо пайплайн штучної трансформацією зображення із 3 вимірною RGB формату у одно каналний формат мозаїки. Для виразності продемонструємо процес на прикладі випадкового зображення із набору MIRFLICKR[35]. Обрано випадкове зображення із індексом 00016 (рис 4.9).



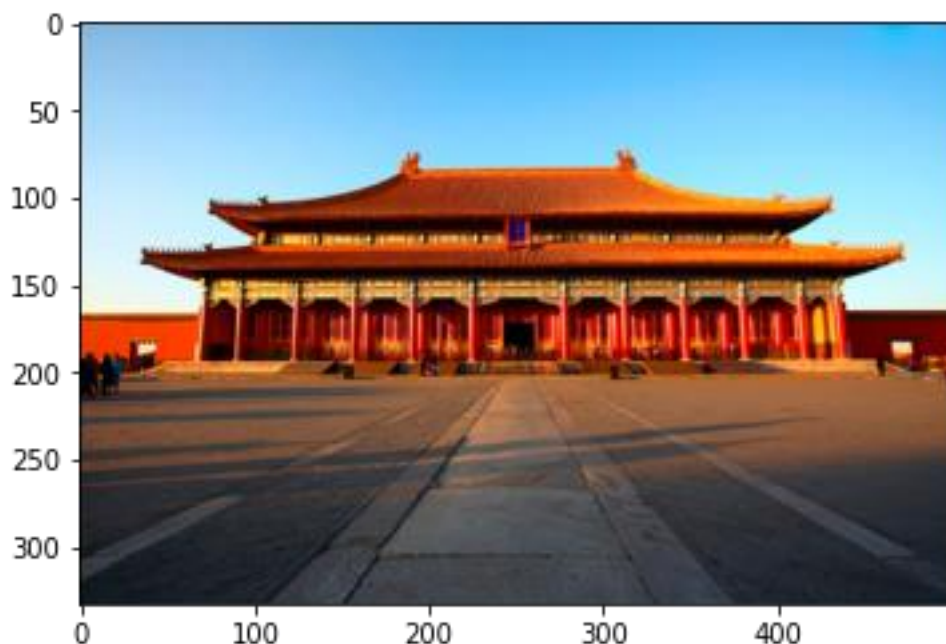


Рисунок 4.9 – Демонстраційне зображення із набору MIRFLICKR

Взявши за основу частину дослідження [31] було вирішено моделювати зворотне відображення тонів(англ. Inverse tone mapping) ,зворотне виправлення кольорів(англ. Inverse color correction) і процес мозаїки.

Ціль тонального відображення лежить у нелінійному перетворенні тональних значень із більш широкого діапазону в більш вузький з метою відтворення зображення на пристрої із меншим динамічним діапазоном.

Оператори тонального відображення поділяються на глобальні та локальні. Глобальні оператори забезпечують перетворення пікселів вхідного зображення на підставі їх інтенсивності і глобальних характеристиках зображення незалежно від положення пікселя. Прикладом глобального типу тонального відображення є тональна крива (тональні криві). Глобальне тональне відображення часто застосовують для обробки даних з 12-бітних сенсорів. При збільшенні динамічного діапазону вхідного зображення переваги глобального перетворення зменшуються, бо глобальне перетворення застосовується до всіх пікселів вхідного зображення однаково, без урахування того, чи знаходяться вони в темній або світлій області, при



цьому оброблені зображення втрачають локальні деталі. Локальні оператори враховують положення пікселя на зображенні, щоб визначити для нього відповідні параметри перетворення. При цьому пікселі рівної інтенсивності після відображення будуть мати різні значення залежно від того, де вони знаходяться: у світлій або темній частині вхідного зображення. Локальне тональний відображення вимагає обрахунку значень оточуючих точок для кожного відображуваного пікселя, що робить процес перетворення досить повільним. В результаті локального перетворення зберігається або навіть збільшується локальний контраст, що, наприклад, дозволяє одночасно зберегти як відтінки білого, так і чорного.

Використовуємо глобальні оператори ,а саме тональні криві що відповідають функціям передачі модуляції із відкритого дослідницького набору DoRF. Цей набір включає 201 приклади функцій ,написаних на мові MatLab .Під час навчання мережі ми будемо випадково вибирати функцію для кожного тренувального зображення. У результаті зміни динамічного діапазону змінюється інтенсивність значень пікселів у динамічно-протилежних областях фотографії(рис 4.10).

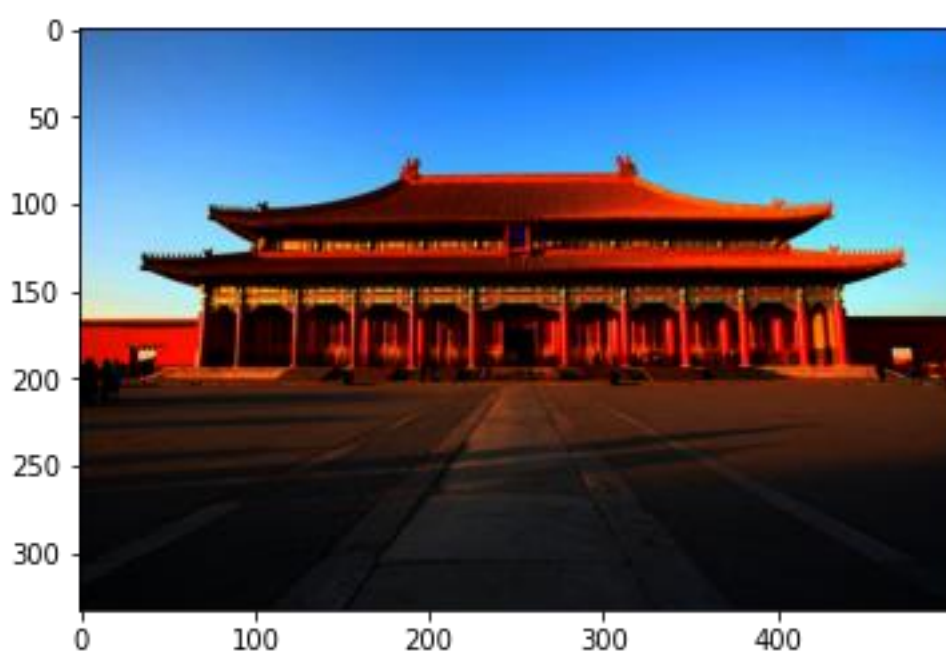


Рисунок 4.10 – Результат зворотного відображення тонів

Наступний крок це перехід із кольорового простору RGB який означає простір відтінків які можна створити з допомогою комбінацій червоного, зеленого і синього кольорів у простір XYZ. Простір XYZ спроектований для уніфікації відтворення візуального сигналу не залежно від типу пристрою (сканер, монітор ,камера) ,оснований на тому як око людини інтерпретує стимул від світла. Це суто математичний простір, у якому основні компоненти є «уявними». Виконуємо трансформацію із використанням метода `rgb2xyz` із бібліотеки `scikits.image.color` (рис. 4.11).

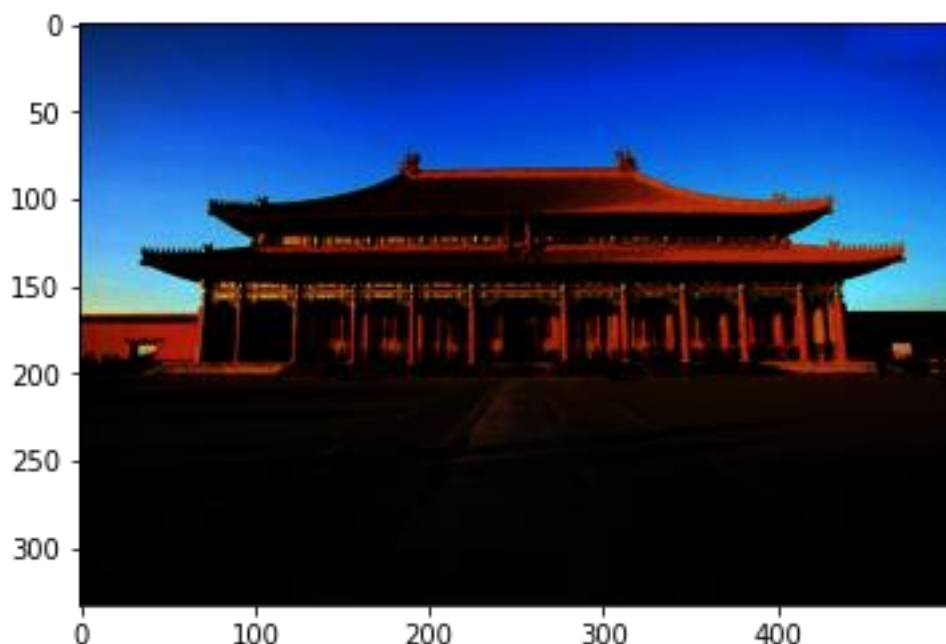


Рисунок 4.11 – Зворотна корекція кольорів

Останній крок для моделювання RAW формату це процес зворотної мозаїки, тобто із трьох каналного зображення вирахувати одновимірне зображення яке відповідає значенню інтенсивності світла у пікселі сенсора накритим фільтром певної довжини .Існує багато видів масивів світлових фільтрів ,для симуляції використовуємо фільтр Байера (4.2) через його простоту і популярність(рис 4.12).

$$CFA = \begin{bmatrix} G & R \\ B & G \end{bmatrix} \quad (4.2)$$

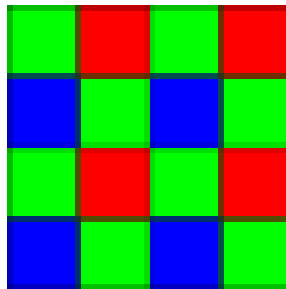


Рисунок 4.12 – Фільтр Байера

Із допомогою зрізу масиву формуємо зображення-мозаїку (рис 4.13).

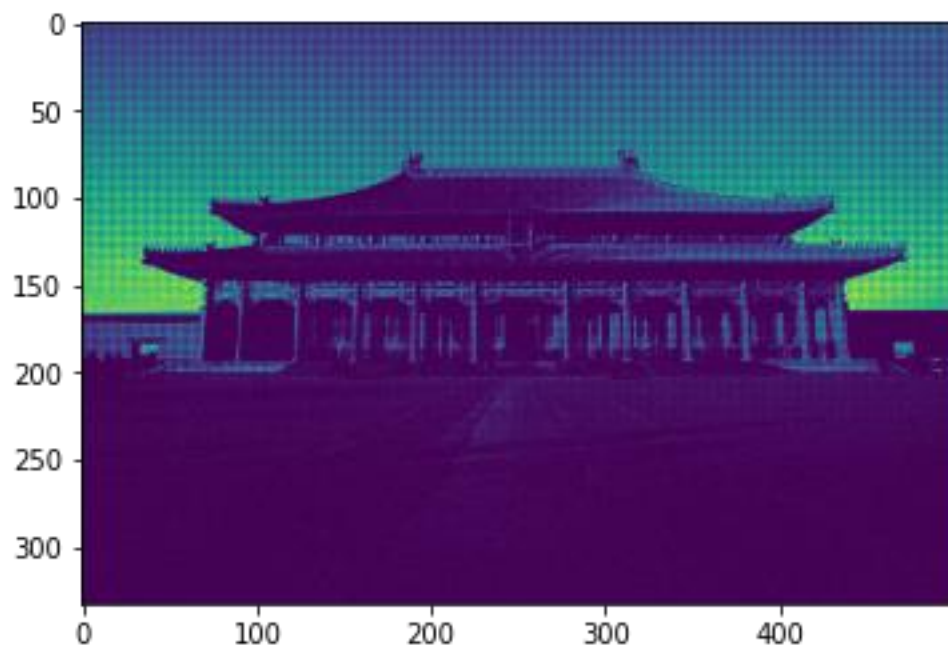


Рисунок 4.13 – Результат моделювання зображення-мозаїки

Виконані операції призвели до створення штучного RAW зображення. Зараз результат зйомки представляє собою зчитані значення інтенсивності світла у пікселі сенсора і їх аналогове-цифрове перетворення. Згідно із проведеним аналізом ,на даному етапі спотворення знімку характеризується в

основному фотонним і електронним шумом. Задаємо шум із двох компонентів як :

$$n(L) = n_p(L) + n_G \quad (4.3)$$

Задаємо розподіл моделі шуму  $n(L) \sim N(0, \sigma^2(L))$ :

$$\sigma^2(L) = L \cdot \sigma_p^2 + \sigma_G^2 \quad (4.4)$$

де  $n_p$  пуасонівська, залежна від сигналу складова із дисперсією  $\sigma_p^2$ ,  $n_G$  складова шуму по Гаусу із дисперсією  $\sigma_G^2$ ,  $L$  опромінення, тобто значення інтенсивності світла у пікселі RAW зображення. Значення  $\sigma_p$  і  $\sigma_G$  рівномірно вибрані із діапазонів  $[0, 0.16]$  і  $[0, 0.06]$  відповідно.

Накладаємо змодельований шум на тестове зображення (рис.4.14).

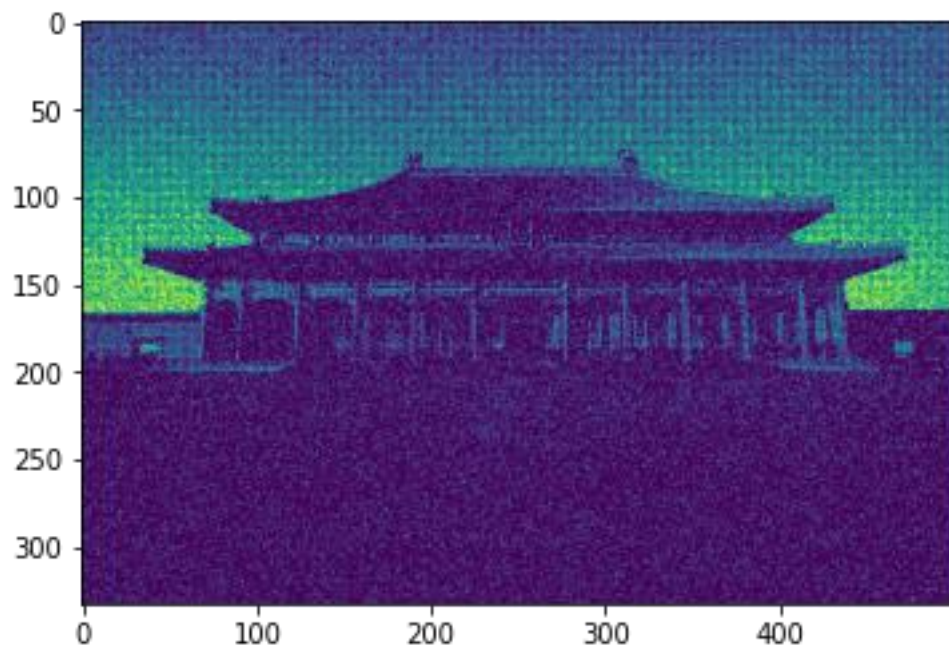


Рисунок 4.14 – RAW зображення із шумом.

Для отриманого зображення виконуємо трансформації ,зворотні до тих які описані вище і утворює результат який буде використовуватись для навчання нейронної мережі (рис 4.15).

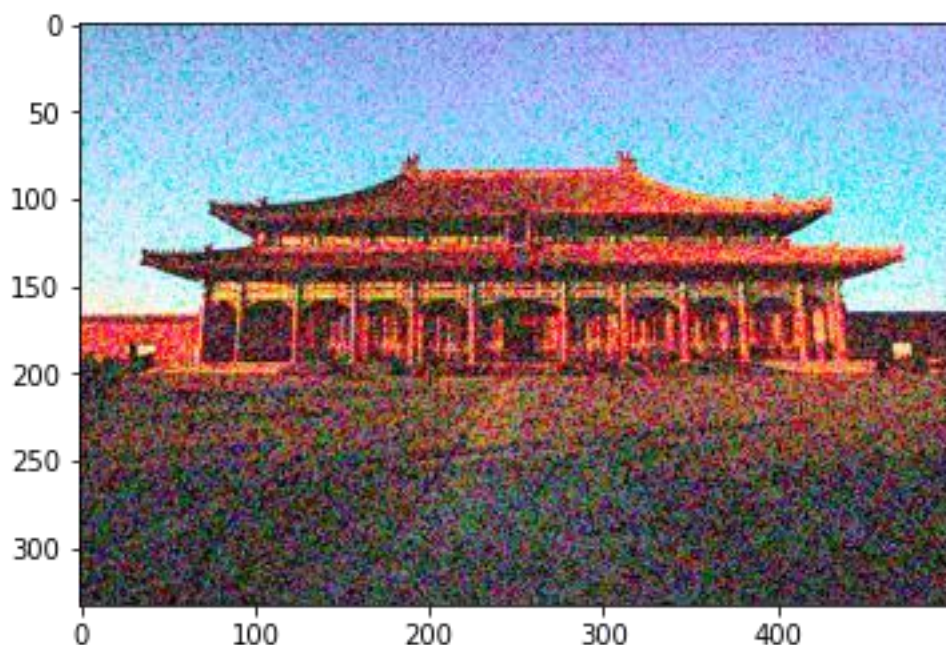


Рисунок 4.15 - Змодельоване зображення ,готове для виконувannya мережі

## 4.2 Розробка структурної схеми

Було вирішено вибрати підхід який ґрунтується спочатку на вимірюванні рівня шуму на зображенні ,отримання так званої карти шумів , а потім не сліпого відновлення. Для отримання оцінки шуму буде використовуватись згорткова нейронна мережа(англ. Convolutional Neural Network, CNN), умовно назвем її  $CNN_n$  ,вибір ґрунтується на можливості цієї мережі відтворити форму входу мережі на її виході. Для несліпого відновлення також буде використовуватись згорткова нейронна мережа

, $CNN_D$ , більш складної будови. На вхід буде сприймати результат  $CNN_n$ -вимірний рівень шуму і зашумлене зображення, на виході – очищене.

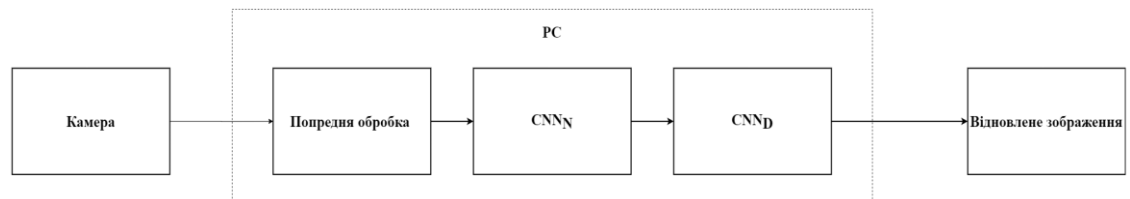


Рисунок 4.16 – Структурна схема системи сліпого відновлення зображень з допомогою нейронних мереж

#### 4.3 Розробка функціональної схеми

Отже, сформулюємо умовну функцію що відповідає отриманню штучного RAW зображення із реалістичними шумами:

$$l = f \left( DM(L + n(L)) \right) \quad (4.5)$$

Де  $l$  позначає штучне зашумлене зображення,  $f(.)$  відповідає за оптичну передаточну функцію із набору DoRF [16],  $DM(.)$  відповідає за функцію демозаїки.  $L$  відповідає за генерацію опромінення світла і дорівнює:

$$L = Mf^{-1}(x) \quad (4.6)$$



де  $M$  функція створення мозаїки,  $x$  чисте зображення.

Використання системи поділяється на два етапи: навчання і використання. На етапі навчання потрібно сформулювати точні вимоги до бажаного рішення і ввести функцію втрат, що буде контролювати мережу під час безпосереднього навчання і вказувати на позитивний чи негативний прогрес, і функцію оцінки моделі яка буде враховувати фінальну продуктивність моделі. Коли продуктивність системи моделі досягає бажаного рівня, навчання припиняється і ваги зберігаються у файл. Все що потрібно для подальшого використання потрібно лише їх загрузка і подання дефектного зображення.

Набір знімків даних для навчання складається із:

- 3000 стиснутих синтетичних зображень із набору MIRFLICKR[35]
- 120 реальних RAW зображень з набору RENOIR[37]

Набір знімків для оцінки роботи системи:

- 10 власних зображень зроблених на камеру смартфона Iphone 7 Plus.

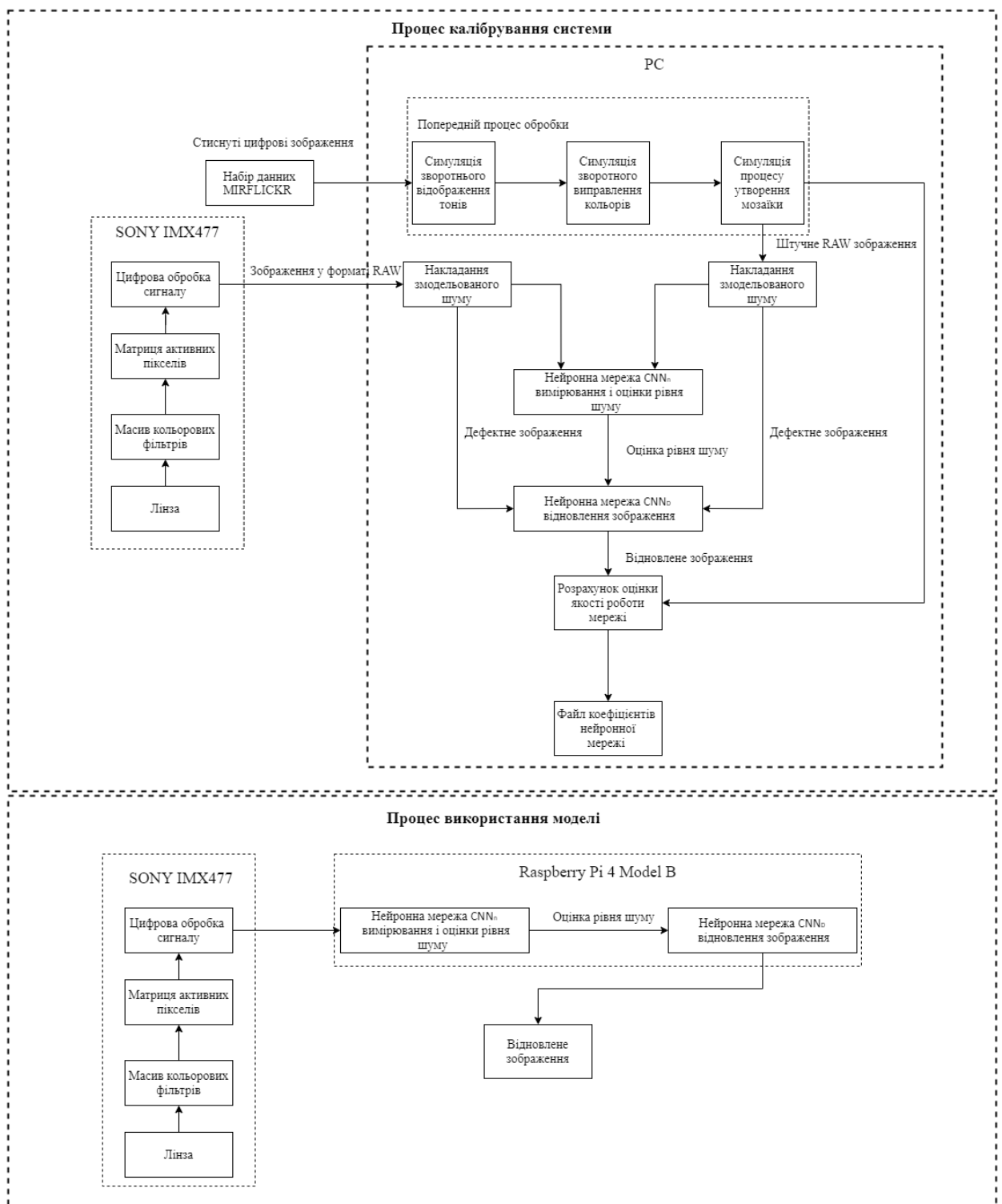


Рисунок 4.17 - Функціональна схема системи сліпого відновлення зображення з допомогою нейронних мереж



#### 4.4 Розробка алгоритму нейронної мережі відновлення зображень

Використовуючи бібліотеку машинного навчання PyTorch і мову програмування Python 3.7.4 було розроблено згорткові нейронні мережі  $CNN_D, CNN_n, CNN_n$  складається із п'яти повністю згорткових шарів і без операцій об'єднання та пакетної нормалізації (рис 4.18). Задача мережі полягає у оцінці рівня шуму дефектного зображення  $l$ :

$$\hat{\sigma}(l) = F(l; w_n) \quad (4.7)$$

де  $w_n$  позначає ваги (параметри) мережі  $CNN_n$ .



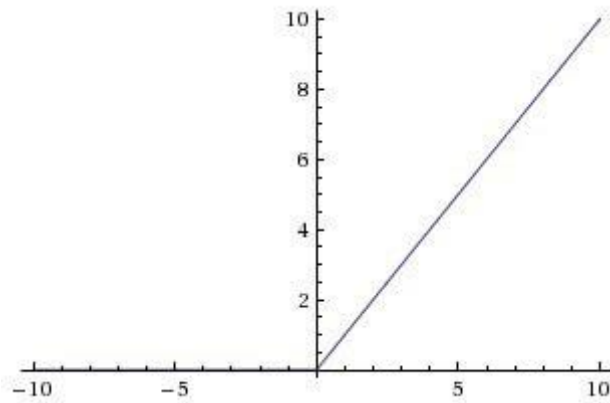


Рисунок 4.19 – Функція активації ReLU

Функція активації визначає вихідне значення нейрона в залежності від результату зваженої суми входів і порогового значення. Іншими словами контролює чи мають зовнішні зв'язки рахувати нейрон як активований, чи його можна ігнорувати. За визначенням нейрон рахується активованим якщо:

$$f(x) = \max(0, x)$$

Тобто, повертає значення  $x$ , якщо  $x > 0$ , і 0 в інших випадках. Позитивні сторони :

- ReLU може бути реалізований за допомогою простого порогового перетворення матриці активації в нулі. Крім того, ReLU не піддається насиченню.
- Використання ReLU фактично підвищує швидкість сходимості стохастичного градієнтного спуску.

За основу мережі  $CNN_D$  взято архітектуру нейронної мережі U-net[33]. U-net – згорткова нейронна мережа, створена для сегментації зображень. Представляє собою повнозв'язну згорткову нейронну мережу. Набула популярності у сфері комп'ютерного зору і використовується на різних типах зображень. Сегментація зображень - це процес розділення цифрового знімку

на декілька частин-сегментів, зміна форми в сторону спрощення для подальшого аналізу, обробки ,візуалізації. В нашому випадку мережа відсегментовує шуми і сцену фотографії використовуючи подану на вхід оцінку (карту) шумів зберігаючи високочастотну інформацію ,тобто по суті виконує задачу локального ітеративного відновлення. Накладання змодельованого шуму призводить до навчання мережі узагальнювати реальні шуми на зображенні. Мережа складається із 16 шарів із симетричним пропуском і жорсткою згорткою, використовується функція активації ReLU. Архітектура мережі  $CNN_D$  представлена у Додатку А. Задача мережі  $CNN_D$  полягає у створенні відновленого зображення  $\hat{l}$  із зашумленого  $l$ :

$$\hat{l} = F_D(l, \hat{\sigma}(l); w_D) \quad (4.8)$$

де  $\hat{\sigma}(l)$  оцінка рівня шуму  $l$ ,  $w_D$  ваги мережі  $CNN_D$ .

Сформуємо функцію втрат що буде оптимізовуватись мережею :

- Введемо похибку реконструкції відновленого зображення  $\hat{l}$  із зашумленого  $l$  на основі евклідової норми:

$$\mathcal{L}_{RE} = \|\hat{l} - l\|_2^2 \quad (4.9)$$

- Недооцінка рівня шуму негативно впливає на результат відновлення тому потрібно ввести асиметричне штрафування :

$$\mathcal{L}_{ASS} = \sum_i |a - I_{ASS}| \cdot (\hat{\sigma}(\hat{l}_i) - \sigma(l_i))^2 \quad (4.10)$$

де  $i$  відповідає за  $i$ -тий піксель зображення,  $a$  коефіцієнт нормалізації,  $I_{ASS}$  умова переоцінки і дорівнює:

$$I_{ASS} = 1 \quad \text{якщо } \hat{\sigma}(\hat{l}_i) - \sigma(l_i) < 0, \text{ або } 0 \text{ в інших випадках} \quad (4.11)$$

Одже, загальна функція втрат :

$$\mathcal{L} = \mathcal{L}_{RE} + \mathcal{L}_{ASS} \quad (4.12)$$

#### 4.5 Деталі впровадження

Для калібрування мереж використовується алгоритм ADAM [26] при  $\beta = 0.9$ . Метод у [18] прийнятий для ініціалізації ваг моделі. Розмір міні-партії - 32, розмір кожної епохи -  $128 \times 128$ . Система моделі підготовлені із 200 епохами, де параметр швидкості навчання для перших 100 епох -  $10^{-3}$ , а потім  $5 \times 10^{-4}$  для подальшого налагодження системи.

#### Висновок

В даній частині дипломної роботи було представлено розробку системи сліпого відновлення зображень із допомогою нейронних мереж. На основі поданих сигналів(цифрових зображень ) ведеться процес калібрації на штучних і реальних наборах даних із подальшим відновленням зображення і вимірюванням оцінки продуктивності системи. Було розроблено структурну, функціональну схеми і схему архітектури нейромереж.

					ВМ61.030004.001 ПЗ	Аркуш
Изм.	Лист	№ докум.	Подпись	Дата		66

## 5 ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ І РОЗРАХУНОК ПОХИБОК

У сфері цифрової обробки сигналів, об'єктивна метрика якості зображення є важливою частиною систем обробки зображень. Одним з важливих застосувань об'єктивних метрик оцінки якості зображень є вимірювання ефективності алгоритмів і систем відновлення зображень. Можливі два підходи до вимірювання якості зображень: кількісна оцінка за допомогою використання математичних методів (среднеквадратическая помилка, заходи, що враховують особливості сприйняття зображення зоровою системою людини) і суб'єктивна оцінка на основі експертних оцінок. Оцінка якості, отримана з використанням існуючих підходів, може значно відрізнятись від оцінки, отриманої за допомогою людей-експертів.

У розробці кількісних оцінок якості зображення досягнуто значних успіхів. Однак введені критерії не є достатньо досконалими. Більшість спроб знайти прийнятні оцінки якості зображення відноситься до окремих випадків. Пропонується якась оцінка, заснована на будь-яких фізіологічних передумовах, а частіше просто зручна для аналізу і обчислень, а потім оцінюються його властивості. Більшість існуючих підходів для оцінки якості використовують еталонне зображення.

Для вимірювання рівня спотворень зображення і оцінки продуктивності роботи системи сліпого відновлення зображень було вирішено використати такі метрики[25]:

- PSNR – пікове співвідношення сигналу до шуму, що означає співвідношення максимального можливого значення сигналу та потужністю шуму, що спотворює сигнал. Ця метрика націлена на кількісну оцінку присутності шуму на зображенні і не описує якісну оцінку процесу відновлення.

- SSIM – (Structure similarity index map) індекс структурної подібності. Метрика обчислює подібність між двома зображеннями звертаючи увагу на порівняння освітлення, контрасту і структурної цілісності представлених зображень. SSIM – індекс являється розвитком методу PSNR і MSE ,які не співвідносяться із фізіологією людського зору .

Використання цих двох методів є загальноприйнятою практикою і використовується у більшості наукових статей і роботах. У даній роботі значення метрик обраховується наступним чином :

$$PSNR = 20 \log_{10} \left[ \frac{255}{RMSE} \right] \quad (5.2)$$

де RMSE корінь із середньоквадратичного відхилення :

$$RMSE = \sqrt{\frac{1}{m \times n} \sum_{i=1}^m \sum_{j=1}^n [I'(i,j) - I(i,j)]^2} \quad (5.3)$$

де  $I', I$  оригінальне зображення і відновлене відповідно,  $m, n$  розмір зображення по вертикалі і горизонталі. Для використання метрик розміри зображень мають співпадати. SSIM розраховують на вікнах  $(x, y)$  розміру  $5 \times 5$  пікселів зображень  $I', I$  відповідно:

$$SSIM(x, y) = \frac{(2\mu_x\mu_y + C_1) \times (2\sigma_{xy} + C_2)}{(\mu_x^2 + \mu_y^2 + C_1) \times (\sigma_x^2 + \sigma_y^2 + C_2)} \quad (5.4)$$

де  $\mu_x$  – середнє  $x$ ,  $\mu_y$  – середнє  $y$ ,  $\sigma_x^2$  – дисперсія  $x$ ,  $\sigma_y^2$  – дисперсія  $y$ ,  $\sigma_{xy}$  коваріація  $x$  і  $y$ ,  $C_1 = (k_1 L_1)^2$ ,  $C_2 = (k_2 L_2)^2$ ,  $L$  динамічний діапазон пікселів.

Обчислимо метрики на 9 тестових зображеннях для оцінки рівня продуктивності моделі. У Таблиці 5.1 вказані значення метрики  $PSNR_N$  що відповідає дефектному зображенню і  $PSNR_R$  позначає відновлене зображення.

Таблиця 5.1 - Вимірювання оцінки відновлення метрикою PSNR

Індекс зображення	Розмір зображення	$PSNR_N$	$PSNR_R$
1	(1380, 1842, 3)	29.36	36.13
2	(1380, 1842, 3)	32.90	35.7
3	(1380, 1842, 3)	29.34	34.01
4	(300, 350, 3)	31.65	35.64
5	(1500, 1500, 3)	27.40	33.78
6	(300, 350, 3)	30.15	34.92
7	(1060, 1105, 3)	33.17	37.78
8	(1060, 1105, 3)	32.99	33.23
9	(1060, 1105, 3)	25.14	28.4

У Таблиці 5.2 вказано вимірювання оцінки відновлення метрикою SSIM.



Таблиця 5.2 - Вимірювання оцінки відновлення метрикою SSIM

Індекс зображення	Розмір зображення	$SSIM_N$	$SSIM_R$
1	(1380, 1842, 3)	0.7622	0.8143
2	(1380, 1842, 3)	0.4574	0.5177
3	(1380, 1842, 3)	0.6170	0.6968
4	(300, 350, 3)	0.7955	0.9052
5	(1500, 1500, 3)	0.8378	0.9477
6	(300, 350, 3)	0.1103	0.1806
7	(1060, 1105, 3)	0.3470	0.4587
8	(1060, 1105, 3)	0.8770	0.9153
9	(1060, 1105, 3)	0.6451	0.7121

### Висновок

Розраховано значення метрик PSNR і SSIM ,як результат оцінки роботи системи.

## 6 РЕЗУЛЬТАТИ ВИКОНАНОЇ РОБОТИ

Тестування роботи системи було проведено із використанням хмарної платформи Google Coolab[39]. Модель навчалась протягом 200 епох і досягла похибку відновлення (4.12) середню на тренувальній вибірці у 0,0021 (рис 6.1) .

...	[203][23]	lr: 0.00000	Loss: 0.0081 (0.0024)	Time: 0.042
	[203][24]	lr: 0.00000	Loss: 0.0011 (0.0023)	Time: 0.076
	[203][25]	lr: 0.00000	Loss: 0.0012 (0.0023)	Time: 0.042
	[203][26]	lr: 0.00000	Loss: 0.0017 (0.0023)	Time: 0.059
	[203][27]	lr: 0.00000	Loss: 0.0026 (0.0023)	Time: 0.039
	[203][28]	lr: 0.00000	Loss: 0.0016 (0.0022)	Time: 0.043
	[203][29]	lr: 0.00000	Loss: 0.0012 (0.0022)	Time: 0.069
	[203][30]	lr: 0.00000	Loss: 0.0008 (0.0022)	Time: 0.058
	[203][31]	lr: 0.00000	Loss: 0.0018 (0.0022)	Time: 0.057
	[203][32]	lr: 0.00000	Loss: 0.0010 (0.0021)	Time: 0.054
	[203][33]	lr: 0.00000	Loss: 0.0021 (0.0021)	Time: 0.047
	[203][34]	lr: 0.00000	Loss: 0.0033 (0.0021)	Time: 0.043
	[203][35]	lr: 0.00000	Loss: 0.0014 (0.0021)	Time: 0.050

Рисунок 6.1 – Процес навчання моделі

На власних зображень ілюструємо результат роботи моделі. Середній час відновлення становить приблизно 4 секунди і залежить від розміру зображень (рис 6.2).

```
Using GPU!
Час обробки: 8.273639765999633 секунд
Час обробки: 1.784129258000121 секунд
Час обробки: 8.22544430500011 секунд
Час обробки: 5.958316721999836 секунд
Час обробки: 4.155256043999998 секунд
Час обробки: 1.4940220449998378 секунд
Час обробки: 0.3879969989998244 секунд
```

Рисунок 6.2 – Час обробки тестових зображень

В цілому система показує хороші результати відновлення які помітні навіть на око. Відновленні зображення не втрачають структурну

будову (ребра не розмитті ,високочастотна інформація збережена)(рис 6.3).

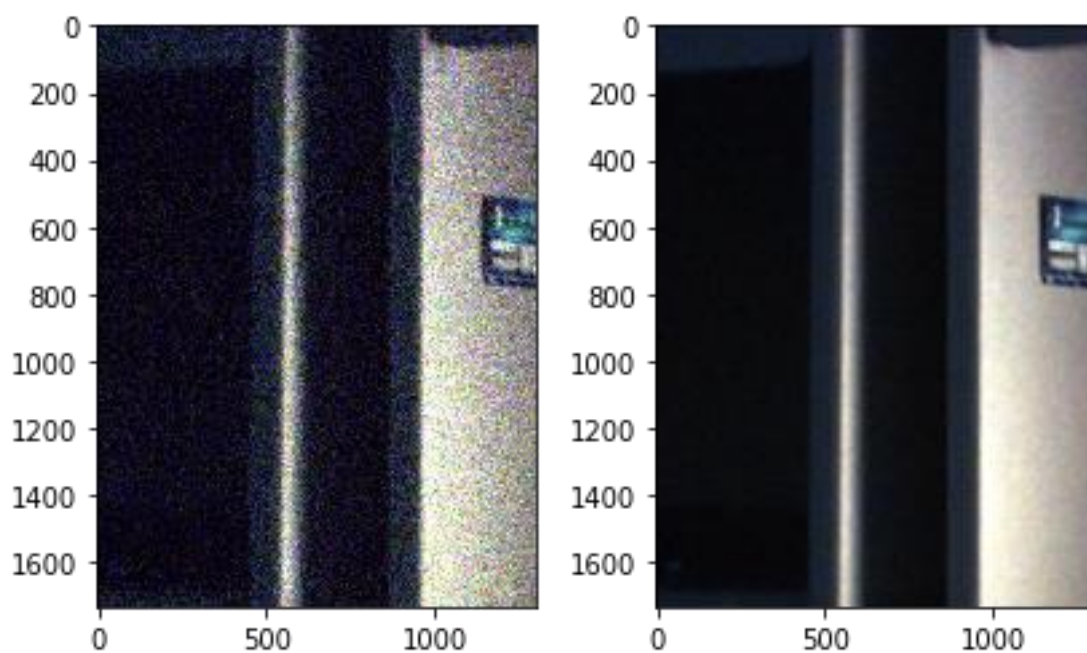


Рисунок 6.3 – Тестове зображення №1

При відновленні зображень близької зйомки наявні незначні артефакти, але в цілому результат задовільний (рис 6.4).

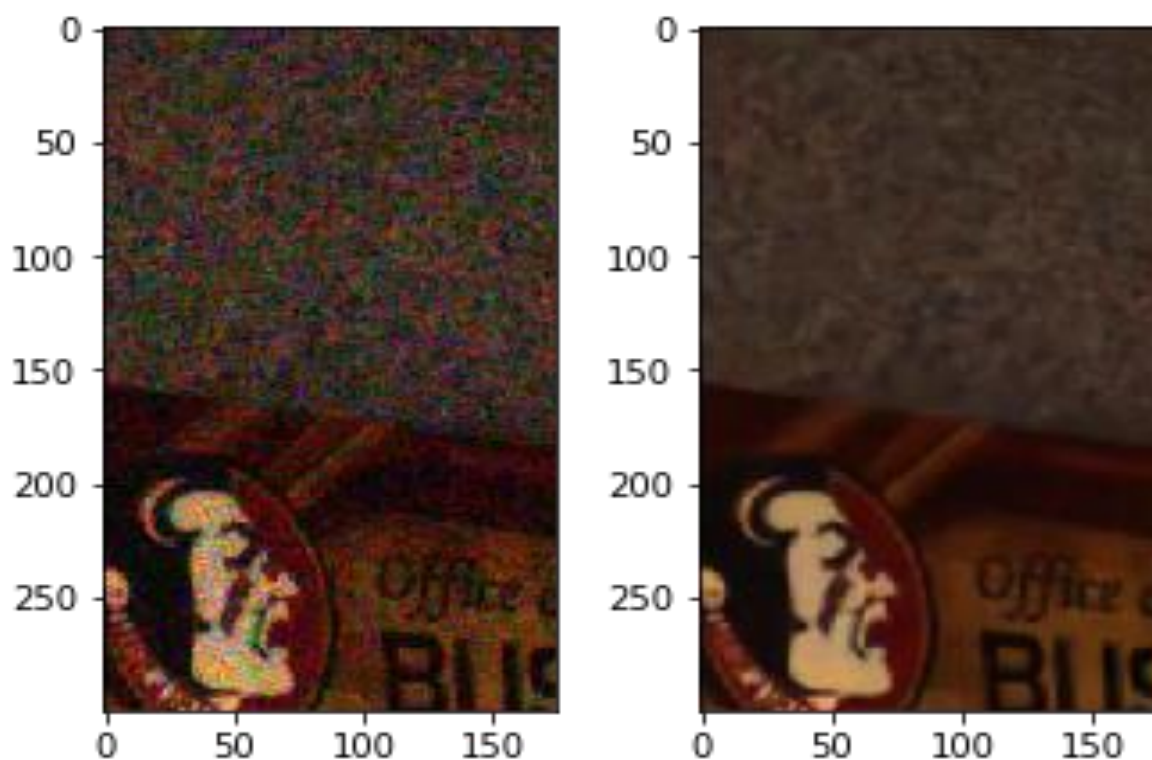


Рисунок 6.4 – Тестове зображення №2

Передача натуральних кольорів зберігається, система не деформує кольоровий простір, розмиття дрібних деталей на зображенні відсутнє (рис 6.5).

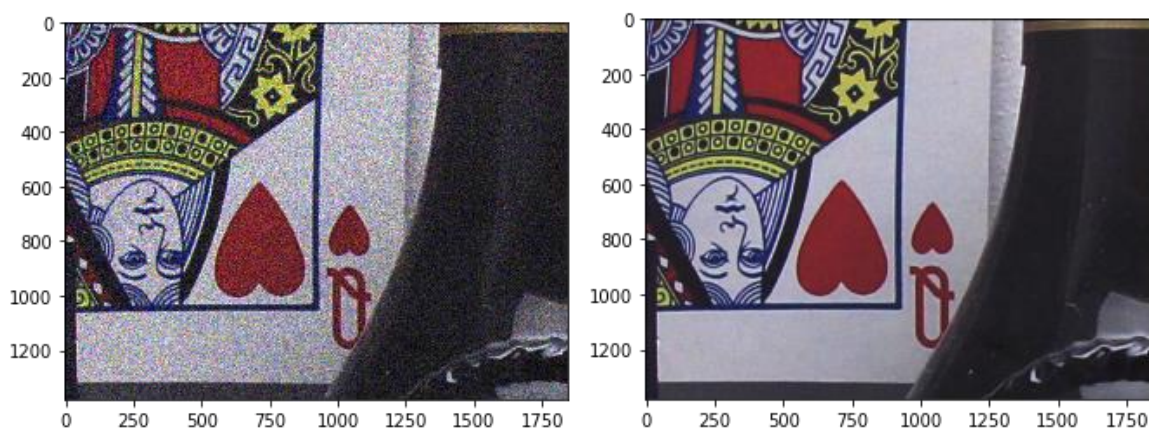


Рисунок 6.5 – Тестове зображення №3

На однотонних зображеннях артефакти відсутні (рис 6.7).

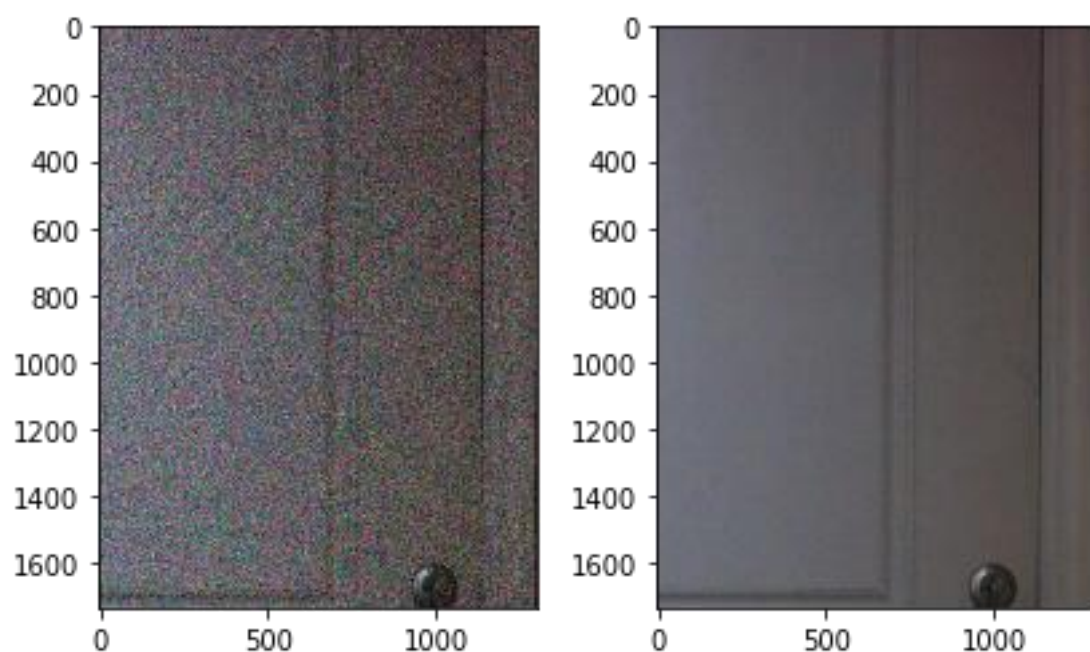


Рисунок 6.7 – Тестове зображення №4

Якщо на зображенні присутній текст ,відновлення підвищує його читаємість (рис 6.8).



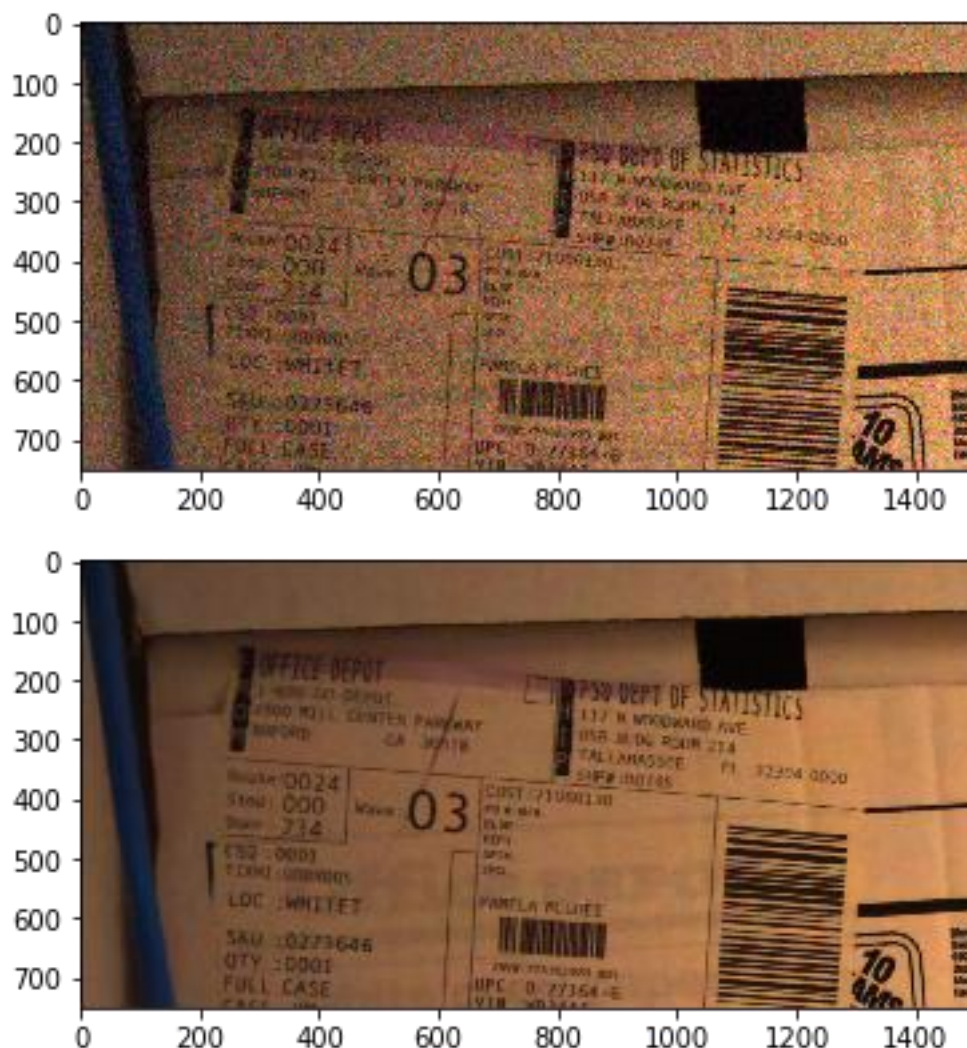


Рисунок 6.8 – Тестове зображення №5

## Висновок

Розроблена система виконує поставлену задачу. Дефектність усувається, деформація зображень не відбувається. Швидкість роботи задовільна і, в разі необхідності, може бути збільшена за рахунок додаткових обчислювальних потужностей.

## ВИСНОВКИ

Даний дипломний проєкт представлений розробленою системою сліпого відновлення зображень із допомогою нейронних мереж. Система виконує наступні функції :

- Симулювання реалістичного шуму сигналів у цифровій обробці зображень.
- Симулювання штучних RAW зображень на основі зжатих .
- Калібрування нейронних мереж згорткової архітектури для сліпого відновлення зображень незалежно від джерела походження.

Для повного опису системи та розгляду її складових , було розроблено такі схеми : структурна ,функціональна і схема алгоритмічна. Також було розроблено технічне завдання, наведено перелік переваг та зроблено аналіз аналогічних методів. Було продемонстровано результат роботи системи для тестових прикладах. Системна справно і задовільно виконує поставлені завдання , тому мета проєкта вважається досягнутою.

					BM61.030004.001 ПЗ	Аркуш
Изм.	Лист	№ докум.	Подпись	Дата		76

## ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Гонсалес Р. Цифровая обработка изображений / Р. Гонсалес, Р. Вудс // – М. : Техносфера, 2005. –1072 с.
2. F. Malgouyres A noise selection approach of image restoration, Applications in signal and image processing IX / F. Malgouyres // Vol 4478. – 2001. – P. 34-41.
3. M. Lindenbaum On Gabor Contribution To Image Enhancement / M. Lindenbaum, M. Fischer and A. M. Bruckstein // Pattern Recognition 27. – 1994. – P. 1-8.
4. L.P. Yaroslavsky Digital Picture Processing. An Introduction / L.P. Yaroslavsky // Berlin-Heidelberg- New York, Springer-Verlag. – 1985. – P. 276.
5. S.M. Smith Susan - a new approach to low level image processing / S.M. Smith and J.M. Brady // International Journal of Computer Vision Vol 23(1). – 1997. – P. 45-78.
6. C. Tomasi Bilateral Filtering for Gray and Color Images / C. Tomasi and R. Manduchi // in Proc. 6th Int. Conf. Computer Vision, New Delhi, India. – 1998.
7. P. Perona Scale space and edge detection using anisotropic diffusion / P. Perona and J. Malik // IEEE Trans. Patt. Anal. Mach. Intell. – 1990. – P. 629-639.
8. L.I. Rudin Nonlinear total variation based noise removal algorithms / L.I. Rudin and E. Fatemi S. Osher // Physica D. – 1992. – P. 259-268.
9. У. К. Прэтт Цифровая обработка изображений / У. К. Прэтт // М.: Мир. – 1982. – P. 523.
10. D. Donoho Ideal spatial adaptation via wavelet shrinkage / D. Donoho, I. Johnstone // Biometrika, vol.81. – 1994. – P. 425–455.
11. D. Donoho Denoising by soft-thresholding / D. Donoho // IEEE Transactions on Information Theory, 41. – 1995. – P. 613–627.



- 12.Y. Wang A Total Variation Wavelet Algorithm for Medical Image Denoising /  
Y. Wang and H. M.Zhou // The International Journal on Biomedical Imaging,  
Volume 2006, article ID 89095. – 2006. – P.6.
- 13.A. Buades Non-Local Algorithm for Image Denoising / A. Buades, B. Coll, and  
J.-M. Morel // In Proceedings of the IEEE Computer Society Conference on  
Computer Vision and Pattern Recognition (CVPR), 2. –2005. – P. 605.
- 14.M. Black Robust Anisotropic Diffusion / M. Black, G. Sapiro, D. Marimont,  
and D. Heeger // IEEE Trans. Image Process., vol. 7, no. 3. – 1998.
- 15.T. F. Chan The Digital TV Filter and Nonlinear Denoising / T. F. Chan, S,  
Osher, and J. Shen // IEEE Trans, Image Proc. – 2001. – P. 231–241.
- 16.A. Efros Texture synthesis by non parametric sampling / A. Efros and T. Leung  
// Proc. Int. Conf.Computer Vision (ICCV 99), Vol. 2, 1991 – P. 1033-1038.
- 17.M. Mahmoudi Fast Image and Video Denoising via Nonlocal Means of Similar  
Neighborhoods / M.Mahmoudi and G. Sapiro// IEEE Signal Processing Letters  
12. – 2005. – P. 79.
- 18.Using Geometry and iterated renement for inverse problems: Total variation  
based image restoration /[S. Osher, M. Burger, D. Goldfarb, J. Xu and W.  
Yin,], CAM-Report 04-13 UCLA. – 2004. – P.47
- 19.Blind Image Deconvolution / [ D. Kundur; D. Hatzinakos] / IEEE Signal  
Processing Magazine, 13 (3), pp. 43-64, 1996.
- 20.Color Image Processing Pipeline / [Rajeev Ramanath, Wesley E. Snyder,  
Youngjun Yoo, and Mark S. Drew ] // IEEE Signal processing magazine [33 -  
43] JANUARY 2005.
- 21.Practical Poissonian-Gaussian noise modeling and fting for single-image raw-  
data /[Alessandro Foi, Mejdi Trimeche, Vladimir Katkovnik, and Karen  
Egiazarian]/ IEEE TRANSACTIONS, TIP-03364-2007-FINAL.
- 22.Different types of noises in images andnoise removing technique / [Mr.Mandar,  
D.Sontakke1, Dr.Mrs. Meghana ,S. Kulkarni] // International Journal of

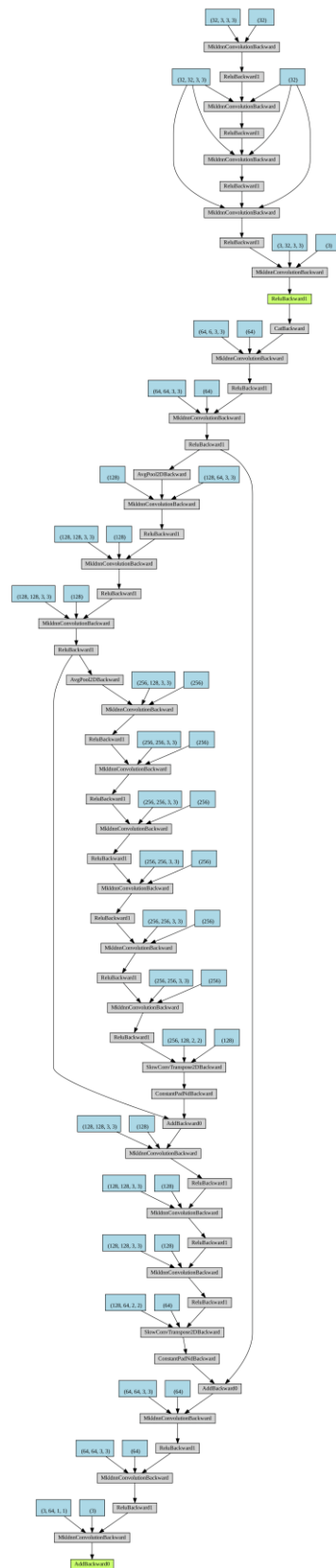
Advanced Technology Engineering and Science No.03, Issue No. 01, January 2015 ISSN (online): 2348 – 7550.

23. Chapter Photon, Poisson noise / [Samuel W. Hasinoff, Google Inc.] // Computer Vision 2014 , ISBN : 978-0-387-30771-8.
24. Realistic camera noise modeling with application to improved HDR synthesis / [Bart Goossens, Hiep Luong, Jan Aelterman, Aleksandra Pizurica and Wilfried Philips] // Article in Journal on Advances in Signal Processing , August 2012, DOI: 10.1186/1687-6180-2012-171.
25. Методика количественной оценки надежности восприятия цифрового изображения / [Е.В. Нариманова, Е.А. Трифонова, А.Е. Килин, М.С. Кучма] // УДК 004.932.2 Informatics and Mathematical Methods in Simulation Vol. 4 (2014), No. 4, pp. 332-336.
26. Blind Image Deconvolution / [D.Kundur; D.Hatzinakos] / IEEE Signal Processing Magazine, 13 (3), pp. 43-64, 1996.
27. A Novel Blind Deconvolution Scheme for Image Restoration using Recursive Filtering / [D. Kundur; D. Hatzinakos] / Journal Article IEEE Transactions on Signal Processing, 46 (2), pp. 375-390, 1998.
28. Toward convolutional blind denoising of real photographs/[ Shi Guo, Zifei Yan, Kai Zhang, Wangmeng Zuo, Lei Zhang, Shi Guo], arXiv:1807.04686v2 [cs.CV] 19 Apr 2019.
29. Image reconstruction using deep learning / [Po-yu Liu] / 3035123887, THE UNIVERSITY OF HONG KONG)
30. Image restoration by blind deconvolution / [Wolfgang Stefan] / 2003/06/27 Technische Universität München (TUM) Arizona State University (ASU) Diploma thesis in techno – mathematics.
31. Unprocessing images for learned raw denoising / [Tim Brooks, Ben Mildenhall, Tianfan Xue] / arXiv:1811.11127v1 [cs.CV] 27 Nov 2018
32. Реставрація зображень методом умовної деконволюції в базисі перетворення RTF / [Шебеста В.] // УДК.621.372.061.

- 33.U-Net: Convolutional Networks for Biomedical Image Segmentation / [Olaf Ronneberger, Philipp Fischer, Thomas Brox] // arXiv:1505.04597 [cs.CV]
- 34.Waterloo Exploration Database: New Challenges for Image Quality Assessment Models[Електронний ресурс].  
URL: <https://ece.uwaterloo.ca/~k29ma/exploration/>
- 35.The MIRFLICKR Retrieval Evaluation [Електронний ресурс]. URL: <https://press.liacs.nl/mirflickr/>
- 36.The Darmstadt Noise Dataset [Електронний ресурс]. URL: <https://noise.visinf.tu-darmstadt.de/>
- 37.RENOIR - A Dataset for Real Low-Light Image Noise Reduction [Електронний ресурс]. URL: <http://ani.stat.fsu.edu/~abarbu/Renoir.html>
- 38.Colaboratory Research Google [Електронний ресурс]. URL: <https://colab.research.google.com>
- 39.Database (DoRF) and Model (EMOR) [Електронний ресурс]. URL: <https://www1.cs.columbia.edu/CAVE//software/softlib/dorf.php>
- 40.АНАЛІЗ МЕТОДІВ ФІЛЬТРАЦІЇ ЗОБРАЖЕНЬ / [К.В. БАРЧЕНКО, С.С. БІЛОШКУРСЬКИЙ, В.В. ГАРМАШ] / Вісник Хмельницького національного університету №4 2012 / УДК 004.932
- 41.Способи організації засобів нейромережевого розпізнавання об'єкта на зображенні / [Буц Вікторія]/ Магістерська дисертація НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ «КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ»

# ДОДАТОК А

## Архітектура нейронної мережі



Изм.	Лист	№ докум.	Подпись	Дата

BM61.030004.001 ПЗ

Аркуш

81

## ДОДАТОК Б

### Лістинг коду програм.

```
import torch

import torch.nn as nn

import torch.nn.functional as F


class CBDNet(nn.Module):

    def __init__(self):
        super(CBDNet, self).__init__()

        self.fcn = FCN()

        self.unet = UNet()

    def forward(self, x):

        noise_level = self.fcn(x)

        concat_img = torch.cat([x, noise_level], dim=1)

        out = self.unet(concat_img) + x

        return noise_level, out


class FCN(nn.Module):

    def __init__(self):
        super(FCN, self).__init__()

        self.inc = nn.Sequential(
            nn.Conv2d(3, 32, 3, padding=1),
            nn.ReLU(inplace=True)
        )

        self.conv = nn.Sequential(
            nn.Conv2d(32, 32, 3, padding=1),
            nn.ReLU(inplace=True)
        )

        self.outc = nn.Sequential(
            nn.Conv2d(32, 3, 3, padding=1),
            nn.ReLU(inplace=True)
        )

    def forward(self, x):

        conv1 = self.inc(x)

        conv2 = self.conv(conv1)
```

					BM61.030004.001 ПЗ	Аркуш
						82
Изм.	Лист	№ докум.	Подпись	Дата		

```

conv3 = self.conv(conv2)
conv4 = self.conv(conv3)
conv5 = self.outc(conv4)
return conv5

```

```

class UNet(nn.Module):
    def __init__(self):
        super(UNet, self).__init__()

        self.inc = nn.Sequential(
            single_conv(6, 64),
            single_conv(64, 64)
        )

        self.down1 = nn.AvgPool2d(2)
        self.conv1 = nn.Sequential(
            single_conv(64, 128),
            single_conv(128, 128),
            single_conv(128, 128)
        )

        self.down2 = nn.AvgPool2d(2)
        self.conv2 = nn.Sequential(
            single_conv(128, 256),
            single_conv(256, 256),
            single_conv(256, 256),
            single_conv(256, 256),
            single_conv(256, 256),
            single_conv(256, 256)
        )

        self.up1 = up(256)
        self.conv3 = nn.Sequential(
            single_conv(128, 128),
            single_conv(128, 128),
            single_conv(128, 128)
        )

```

```

        self.up2 = up(128)

        self.conv4 = nn.Sequential(
            single_conv(64, 64),
            single_conv(64, 64)
        )

        self.outc = outconv(64, 3)

    def forward(self, x):
        inx = self.inc(x)

        down1 = self.down1(inx)
        conv1 = self.conv1(down1)

        down2 = self.down2(conv1)
        conv2 = self.conv2(down2)

        up1 = self.up1(conv2, conv1)
        conv3 = self.conv3(up1)

        up2 = self.up2(conv3, inx)
        conv4 = self.conv4(up2)

        out = self.outc(conv4)
        return out

class single_conv(nn.Module):
    def __init__(self, in_ch, out_ch):
        super(single_conv, self).__init__()
        self.conv = nn.Sequential(
            nn.Conv2d(in_ch, out_ch, 3, padding=1),
            nn.ReLU(inplace=True)
        )

    def forward(self, x):
        x = self.conv(x)
        return x

class up(nn.Module):

```

```

def __init__(self, in_ch):
    super(up, self).__init__()
    self.up = nn.ConvTranspose2d(in_ch, in_ch//2, 2, stride=2)

def forward(self, x1, x2):
    x1 = self.up(x1)

    # input is CHW
    diffY = x2.size()[2] - x1.size()[2]
    diffX = x2.size()[3] - x1.size()[3]

    x1 = F.pad(x1, (diffX // 2, diffX - diffX//2,
                    diffY // 2, diffY - diffY//2))

    x = x2 + x1
    return x

class outconv(nn.Module):
    def __init__(self, in_ch, out_ch):
        super(outconv, self).__init__()
        self.conv = nn.Conv2d(in_ch, out_ch, 1)

    def forward(self, x):
        x = self.conv(x)
        return x

class fixed_loss(nn.Module):
    def __init__(self):
        super().__init__()

    def forward(self, out_image, gt_image, est_noise, gt_noise, if_asym):
        h_x = est_noise.size()[2]
        w_x = est_noise.size()[3]
        count_h = self._tensor_size(est_noise[:, :, 1:, :])
        count_w = self._tensor_size(est_noise[:, :, :, 1:])
        h_tv = torch.pow((est_noise[:, :, 1:, :] - est_noise[:, :, :h_x-1, :]),
2).sum()
        w_tv = torch.pow((est_noise[:, :, :, 1:] - est_noise[:, :, :, :w_x-1]),
2).sum()
        tvloss = h_tv / count_h + w_tv / count_w

        loss = torch.mean(torch.pow((out_image - gt_image), 2)) + \

```



```

        if_asym * 0.5 * torch.mean(torch.mul(torch.abs(0.3 - F.relu(gt_noise -
est_noise))), torch.pow(est_noise - gt_noise, 2))) + \

        0.05 * tvloss

    return loss

def _tensor_size(self,t):

    return t.size()[1]*t.size()[2]*t.size()[3]

```

## Алгоритм навчання моделей :

```

import numpy as np

import cv2

import os

import math

import scipy.io as sio

import matplotlib.pyplot as plt

from scipy.optimize import curve_fit

class AverageMeter(object):

    def __init__(self):

        self.reset()

    def reset(self):

        self.val = 0

        self.avg = 0

        self.sum = 0

        self.count = 0

    def update(self, val, n=1):

        self.val = val

        self.sum += val * n

        self.count += n

        self.avg = self.sum / self.count

def ReadImg(filename):

    img = cv2.imread(filename)

    img = img[:,:,:-1] / 255.0

    img = np.array(img).astype('float32')

    return img

def hwc_to_chw(img):

```

					BM61.030004.001 ПЗ	Аркуш
						86
Изм.	Лист	№ докум.	Подпись	Дата		

```

        return np.transpose(img, axes=[2, 0, 1])

def chw_to_hwc(img):
    return np.transpose(img, axes=[1, 2, 0])

##### noise model #####

def func(x, a):
    return np.power(x, a)

def CRF_curve_fit(I, B):
    popt, pcov = curve_fit(func, I, B)
    return popt

def CRF_function_transfer(x, y):
    para = []
    for crf in range(201):
        temp_x = np.array(x[crf, :])
        temp_y = np.array(y[crf, :])
        para.append(CRF_curve_fit(temp_x, temp_y))
    return para

def mosaic_bayer(rgb, pattern, noiselevel):

    w, h, c = rgb.shape
    if pattern == 1:
        num = [1, 2, 0, 1]
    elif pattern == 2:
        num = [1, 0, 2, 1]
    elif pattern == 3:
        num = [2, 1, 1, 0]
    elif pattern == 4:
        num = [0, 1, 1, 2]
    elif pattern == 5:
        return rgb

    mosaic = np.zeros((w, h, 3))
    mask = np.zeros((w, h, 3))
    B = np.zeros((w, h))

```

```

B[0:w:2, 0:h:2] = rgb[0:w:2, 0:h:2, num[0]]
B[0:w:2, 1:h:2] = rgb[0:w:2, 1:h:2, num[1]]
B[1:w:2, 0:h:2] = rgb[1:w:2, 0:h:2, num[2]]
B[1:w:2, 1:h:2] = rgb[1:w:2, 1:h:2, num[3]]

gauss = np.random.normal(0, noiselevel/255., (w, h))
gauss = gauss.reshape(w, h)
B = B + gauss

return (B, mask, mosaic)

def ICRF_Map(Img, I, B):
    w, h, c = Img.shape
    output_Img = Img.copy()
    prebin = I.shape[0]
    tiny_bin = 9.7656e-04
    min_tiny_bin = 0.0039
    for i in range(w):
        for j in range(h):
            for k in range(c):
                temp = output_Img[i, j, k]
                start_bin = 0
                if temp > min_tiny_bin:
                    start_bin = math.floor(temp/tiny_bin - 1) - 1
                for b in range(start_bin, prebin):
                    tempB = B[b]
                    if tempB >= temp:
                        index = b
                        if index > 0:
                            comp1 = tempB - temp
                            comp2 = temp - B[index-1]
                            if comp2 < comp1:
                                index = index - 1
                        output_Img[i, j, k] = I[index]
                    break

    return output_Img

```

					BM61.030004.001 ПЗ	Аркуш
						88
Изм.	Лист	№ докум.	Подпись	Дата		

```

def CRF_Map(Img, I, B):
    w, h, c = Img.shape
    output_Img = Img.copy()
    prebin = I.shape[0]
    tiny_bin = 9.7656e-04
    min_tiny_bin = 0.0039
    for i in range(w):
        for j in range(h):
            for k in range(c):
                temp = output_Img[i, j, k]

                if temp < 0:
                    temp = 0
                    Img[i, j, k] = 0
                elif temp > 1:
                    temp = 1
                    Img[i, j, k] = 1
                start_bin = 0
                if temp > min_tiny_bin:
                    start_bin = math.floor(temp/tiny_bin - 1) - 1

                for b in range(start_bin, prebin):
                    tempB = I[b]
                    if tempB >= temp:
                        index = b
                        if index > 0:
                            comp1 = tempB - temp
                            comp2 = temp - B[index-1]
                            if comp2 < comp1:
                                index = index - 1
                        output_Img[i, j, k] = B[index]
                    break

    return output_Img

def CRF_Map_opt(Img, popt):
    w, h, c = Img.shape
    output_Img = Img.copy()

    output_Img = func(output_Img, *popt)

```

```

        return output_Img

def Demosaic(B_b, pattern):

    B_b = B_b * 255
    B_b = B_b.astype(np.uint16)

    if pattern == 1:
        lin_rgb = cv2.demosaicing(B_b, cv2.COLOR_BayerGB2BGR)
    elif pattern == 2:
        lin_rgb = cv2.demosaicing(B_b, cv2.COLOR_BayerGR2BGR)
    elif pattern == 3:
        lin_rgb = cv2.demosaicing(B_b, cv2.COLOR_BayerBG2BGR)
    elif pattern == 4:
        lin_rgb = cv2.demosaicing(B_b, cv2.COLOR_BayerRG2BGR)
    elif pattern == 5:
        lin_rgb = B_b

    lin_rgb = lin_rgb[:, :, ::-1] / 255.
    return lin_rgb

def AddNoiseMosai(x, CRF_para, iCRF_para, I, B, Iinv, Binv, sigma_s, sigma_c,
crf_index, pattern, opt = 1):
    w, h, c = x.shape
    temp_x = CRF_Map_opt(x, iCRF_para[crf_index] )

    sigma_s = np.reshape(sigma_s, (1, 1, c))
    noise_s_map = np.multiply(sigma_s, temp_x)
    noise_s = np.random.randn(w, h, c) * noise_s_map
    temp_x_n = temp_x + noise_s

    noise_c = np.zeros((w, h, c))
    for chn in range(3):
        noise_c[:, :, chn] = np.random.normal(0, sigma_c[chn], (w, h))

    temp_x_n = temp_x_n + noise_c
    temp_x_n = np.clip(temp_x_n, 0.0, 1.0)
    temp_x_n = CRF_Map_opt(temp_x_n, CRF_para[crf_index])

    if opt == 1:

```

					BM61.030004.001 ПЗ	Аркуш
						90
Изм.	Лист	№ докум.	Подпись	Дата		

```

        temp_x = CRF_Map_opt(temp_x, CRF_para[crf_index])

    B_b_n = mosaic_bayer(temp_x_n[:, :, ::-1], pattern, 0)[0]
    lin_rgb_n = Demosaic(B_b_n, pattern)
    result = lin_rgb_n

    if opt == 1:
        B_b = mosaic_bayer(temp_x[:, :, ::-1], pattern, 0)[0]
        lin_rgb = Demosaic(B_b, pattern)
        diff = lin_rgb_n - lin_rgb
        result = x + diff

    return result

def AddRealNoise(image, CRF_para, iCRF_para, I_gl, B_gl, I_inv_gl, B_inv_gl):
    sigma_s = np.random.uniform(0.0, 0.16, (3,))
    sigma_c = np.random.uniform(0.0, 0.06, (3,))
    CRF_index = np.random.choice(201)
    pattern = np.random.choice(4) + 1

    noise_img = AddNoiseMosai(image, CRF_para, iCRF_para, I_gl, B_gl, I_inv_gl,
                                B_inv_gl, sigma_s, sigma_c, CRF_index, pattern, 0)
    noise_level = sigma_s * np.power(image, 0.5) + sigma_c

    return noise_img, noise_level

```

## Вибір параметрів запуску і відновлення зображень.

```

from __future__ import division
from __future__ import print_function
import os, time, scipy.io, shutil
import argparse
import torch
import torch.nn as nn
import torch.nn.functional as F
import numpy as np
import glob
import re
import cv2
from PIL import Image
from SSIM import *
import matplotlib.pyplot as plt
from utils import *

```

					ВМ61.030004.001 ПЗ	Аркуш
						91
Изм.	Лист	№ докум.	Подпись	Дата		

```

from model import *

parser = argparse.ArgumentParser(description='Testing on DND dataset')
parser.add_argument('--ckpt', type=str, default='all',
                    choices=['all', 'real', 'synthetic'], help='checkpoint type')
parser.add_argument('--cpu', nargs='?', const=1, help = 'Use CPU')
args = parser.parse_args()

input_dir = './d/some_test/'
#input_dir = './d/test/'
checkpoint_dir = './checkpoint/' + args.ckpt + '/'
result_dir = './result/'

test_fns = glob.glob(input_dir + '*.bmp')
#test_fns = glob.glob(input_dir + '0001.bmp')
#test_fns.append(*glob.glob(input_dir + '0002.bmp'))
print(test_fns)

# model load
if os.path.exists(checkpoint_dir + 'checkpoint.pth.tar'):
    # load existing model
    model_info = torch.load(checkpoint_dir + 'checkpoint.pth.tar')
    print('==> loading existing model:', checkpoint_dir + 'checkpoint.pth.tar')
    model = CBDNet()
    if not args.cpu:
        print('Using GPU!')
        model.cuda()
    else:
        print('Using CPU!')
    model.load_state_dict(model_info['state_dict'])
else:
    print('Error: No trained model detected!')
    exit(1)

if not os.path.isdir(result_dir + 'test/'):
    os.makedirs(result_dir + 'test/')

dtype = torch.cuda.FloatTensor
ssim = SSIM().type(dtype)

for ind, test_fn in enumerate(test_fns):

```

					BM61.030004.001 ПЗ	Архив
Изм.	Лист	№ докум.	Подпись	Дата		92

```

model.eval()

with torch.no_grad():
    print(test_fn)
    noisy_img = cv2.imread(test_fn)
    shape = noisy_img.shape
    noisy_img = noisy_img[:int(shape[0]/2),:int(shape[1]/2),:]
    noisy_img = noisy_img[:,:,:-1] / 255.0
    noisy_img = np.array(noisy_img).astype('float32')

    temp_noisy_img = noisy_img
    temp_noisy_img_chw = hwc_to_chw(temp_noisy_img)

    input_var =
    torch.from_numpy(temp_noisy_img_chw.copy()).type(torch.FloatTensor).unsqueeze(0)

    if not args.cpu:
        input_var = input_var.cuda()
    _, output = model(input_var)

    output_np = output.squeeze().cpu().detach().numpy()
    output_np = chw_to_hwc(np.clip(output_np, 0, 1))

    ar = Image.fromarray((temp_noisy_img * 255).astype(np.uint8))
    ar1 = Image.fromarray((output_np * 255).astype(np.uint8))

    ar.save(result_dir + 'test/test_%d.jpg'%(ind))
    ar1.save(result_dir + 'test/test_ready_%d.jpg'%(ind))
    #plt.imshow(temp)

    #print(ssim((output_np *
    255).astype(np.uint8), (temp_noisy_img *
    255).astype(np.uint8)))

```